

## THESIS / THÈSE

### MASTER EN SCIENCES MATHÉMATIQUES

#### Apport du graphique à l'optimisation

#### Définition d'une méthodologie applicable tant dans le domaine industriel que celui de l'enseignement

HALLET, Etienne

*Award date:*  
1986

*Awarding institution:*  
Université de Namur

[Link to publication](#)

#### General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

#### Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

FACULTES UNIVERSITAIRES N.D DE LA PAIX  
NAMUR  
FACULTE DES SCIENCES

**Apport du graphique à  
l'optimisation**

Définition d'une méthodologie applicable  
tant dans le domaine industrielle que celui  
de l'enseignement

Mémoire présenté pour l'obtention du grade  
de Licencié en Sciences  
mathématiques  
par

Promoteur :

**Hallet Etienne**

**NGUYEN VAN HIEN**

ANNEE ACADEMIQUE : 1985-1986



Au'il nous soit permis de remercier  
tous ceux qui ont contribué à l'aboutissement  
de ce mémoire.

Nous remercions plus particulièrement,  
Monsieur Verginon, chargé du centre de calcul  
de la S.A. Solvay, pour l'attention constante  
qu'il a accordée à notre recherche et pour  
l'aide précieuse qu'il a apportée à la rédaction  
de ce mémoire.

Notre reconnaissance va également à  
notre promoteur de mémoire, Monsieur Nguyen  
Van Hien, qui nous a prodigué ses conseils  
et ses encouragements.

*effort*

## TABLE DES MATIERES.

---

	PAGE.
0. INTRODUCTION	1.
1. POSITION DU PROBLEME	5.
2. OPTIMISATION D'UNE FONCTION REELLE D'UNE VARIABLE REELLE.	8.
2.0. Introduction	8.
2.1. Méthode de Newton.	10.
2.1.1. Description de l'algorithme	10.
2.1.2. Implémentation du graphique	14.
2.1.3. Exemples	19.
2.2. Méthode par interpolation cubique	21.
2.2.1. Description de l'algorithme.	21.
2.2.2. Implémentation du graphisme.	32.
2.2.3. Exemple	34.
2.3. Conclusions.	35.
3. OPTIMISATION D'UNE FONCTION REELLE DE PLUSIEURS VARIABLES REELLES.	36.
3.0. Introduction	36.
3.1. Méthode Quasi-Newton pour résoudre les problèmes d'optimisation sans contrainte .	37.

3.1.0. Introduction	37.
3.1.1. Description de l'algorithme de Broyden, Fletcher, Goldfarb, Shanno.	41.
3.1.2. Implémentation du graphique	59.
3.1.3. Exemples.	62.
3.2. Méthode pour résoudre des problèmes d'optimisation avec contrainte de bornes	65.
3.2.0. Introduction	65.
3.2.0.1. Stratégie des contraintes actives pour résoudre les problèmes d'optimisation avec contraintes de bornes.	66.
3.2.02 Adaptation de la stratégie des contraintes actives aux problèmes avec con- traintes de bornes unique- ment.	70.
3.2.1. Structure d'implémentation et résolution du problème selon NAG.	78.
3.2.2. Implémentation du graphis- me.	118.
3.2.3. Exemple.	119.
3.3. Conclusions générales.	120.

4. INFORMATISATION DU PROBLEME.	121.
4.0. Introduction	121.
4.1. Outils et moyens disponibles.	123.
4.2. Méthodologie générale.	127.
4.3. Le logiciel OPTIGRAPH.	129.
5. APPLICATION A UN PROBLEME DE CINETIQUE CHIMIQUE	155.
5.0. Position du problème.	155.
5.1. Résultats du calcul.	161.
6. CONCLUSIONS GENERALES.	163.
BIBLIOGRAPHIE	166.
Annexe 1 : Factorisation de Cholesky ( théorèmes fondamentaux )	168.
Annexe 2 : Logiciel OPTIGRAPH	188.
Annexe 3 : programme QNVISU	245.
Annexe 4 : code Newton	301.
Annexe 5 : code Cubique	319.

# **Introduction**

## INTRODUCTION.

---

Le présent travail a pour objectif d'étudier dans quelle mesure l'outil graphique peut venir en aide à l'optimisation mathématique d'une fonction réelle d'une ou de plusieurs variables réelles.

Les développements récents de l'informatique ont entre autres débouché sur des perspectives intéressantes en matière de graphique, d'une part grâce à l'introduction d'une norme affranchissant la conception et le développement de programmes des contraintes liées au matériel graphique utilisé et d'autre part grâce aux progrès technologiques dans ce domaine.

D'où l'idée d'utiliser ces outils comme moyen d'aide à l'optimisation à trois stades différents du processus :

- 1- au démarrage, c'est-à-dire dans le choix du point de départ;
- 2- pendant la procédure d'optimisation par un suivi des différents itérés;
- 3- à l'arrêt des calculs, au point supposé être le minimum de la fonction.

L'aide au démarrage est particulièrement utile dans les problèmes rencontrés dans la recherche industrielle

où les fonctions à minimiser sont très souvent multimodales, situation parfois compliquée par les corrélations existant entre variables conduisant à des extrémums "plats".

Le suivi de la procédure itérative est important dans la mesure où les algorithmes ne tenant pas compte de la nature physique du problème peuvent s'égarer vers des extrémums locaux sans intérêt.

En outre, le contrôle de la progression du point courant s'avère très utile pour comparer les performances d'algorithmes différents pour un problème donné. Une telle démarche est surtout intéressante dans les problèmes de minimisation et d'identification qui nécessitent des calculs de fonction coûteux.

Enfin, le contrôle de l'acquisition du minimum par des moyens graphiques est un complément utile aux critères mathématiques classiques surtout dans le cas où certaines variables sont fortement corrélées.

Nous avons divisé ce travail en six chapitres.

- Le premier pose le problème et donne les limites du travail.

- Le deuxième est consacré à l'optimisation d'une fonction réelle d'une variable réelle.

Deux méthodes classiques sont analysées



( la méthode de Newton et la méthode par interpolation cubique); elles permettent de situer les points d'ancrage entre l'algorithme mathématique et le graphisme.

Des exemples illustrent ces développements théoriques.

Le paragraphe des conclusions reprend les enseignements des premiers essais.

- Le troisième chapitre est consacré à l'optimisation d'une fonction réelle de plusieurs variables réelles. Nous y considérons à la fois le problème sans et avec contraintes de bornes.

Les algorithmes choisis pour cette étude (les méthodes B.F.G.S et B.F.G.S. appliquées dans la stratégie des contraintes actives) ont été étudiés en détail de manière à en tirer un maximum d'informations utiles pour les visualisations.

L'implémentation du graphisme est décrite dans chacun des cas et des exemples complètent les développements théoriques. Ici aussi, un paragraphe de conclusion rassemble les résultats de nos premières expériences.

- Le chapitre 4 traite de l'informatisation du problème. Un logiciel OPTIGRAPH a été développé au centre de calcul du laboratoire de recherches de SOLVAY S.A. OPTIGRAPH est conçu dans une optique conviviale et portable.

La convivialité est une condition nécessaire pour que les chercheurs de différentes disciplines puissent y avoir accès



sans devoir subir une formation rebutante.

La portabilité est une exigence dans un groupe industriel qui dispose de matériels informatiques différents : les programmes doivent pouvoir s'adapter d'un système à l'autre avec un minimum d'effort. Nous avons dans cette optique utilisé un maximum d'outils standardisés de développement .

Le chapitre 5 est, quant à lui, consacré à une application de la méthodologie que nous proposons, à un problème industriel.

Nous y mettons en évidence les avantages et les limitations de notre approche.

Le sixième et dernier chapitre reprend les conclusions générales.

# Chapitre 1

## 1. POSITION DU PROBLEME.

De nombreuses applications dans le domaine de la recherche industrielle requièrent la minimisation d'une fonction réelle de plusieurs variables réelles. Aussi bien les modèles déterministes que statistiques conduisent-ils à de telles formulations.

En particulier, l'identification paramétrique, qui consiste à rechercher les paramètres d'un modèle en minimisant l'écart entre les valeurs théoriques calculées par ce dernier et les résultats expérimentaux, nous amène fréquemment à devoir optimiser une fonctionnelle de type moindres carrés ( $L^2$ ) ou valeur absolue ( $L^1$ ).

Le nombre de paramètres à ajuster n'excède en général pas la vingtaine.

Les difficultés rencontrées dans la résolution de tels problèmes sont de plusieurs ordres :

- tout d'abord, les modèles étudiés évoluent en fonction de l'état d'avancement de la recherche. Pratiquement, cela se traduit par des adaptations de la fonctionnelle à optimiser qui modifient parfois complètement son allure et la complexité de son calcul.

D'où la nécessité de disposer, à priori, d'algorithmes flexibles et robustes.

- Ensuite, on ne dispose pas toujours d'une estimation réaliste de chacun des paramètres à ajuster lorsqu'on entame la procédure de minimisation.  
Il peut arriver que, pour certains d'entre eux, on n'ait que de très vagues ordres de grandeur.
- Enfin, il arrive fréquemment, c'est le cas dans les cinétiques chimiques par exemple, que les paramètres soient fortement corrélés ce qui ajouté aux imprécisions sur les estimations initiales peut conduire à des résultats physiquement aberrants.

Si l'on ajoute à cela le fait que les algorithmes mathématiques recherchent un extrémum local sans se préoccuper du sens physique des variables, on se trouve contraint à utiliser, dans les cas difficiles, une technique d'essais et erreurs particulièrement coûteuse en ressources humaines et machines.

C'est pourquoi on a pensé coupler l'outil graphique, devenu aujourd'hui d'une grande flexibilité, à l'algorithme mathématique de minimisation dans le but de faciliter la recherche de l'optimum d'une fonction de plusieurs variables.

L'aide du graphique doit être pensé sous trois formes :

- Aide au démarrage de la procédure de minimisation c'est-à-dire au choix des valeurs initiales des paramètres;

- Suivi du point courant pendant la recherche de l'optimum et prise de décision éventuelle par le chercheur;
- Contrôle des résultats à la fin de la procédure d'optimisation.

Le premier point peut être résolu par des visualisations ( courbes de niveaux et/ou vues isométriques ), le deuxième le sera par des coupes dans des plans privilégiés ( plan des directions de descente, plan formé par les vecteurs propres correspondants au plus grandes valeurs propres du hessien de la fonction à minimiser si celle-ci est suffisamment différentiable), cette analyse pouvant déjà conduire à une modification de la stratégie d'optimisation. Le contrôle des résultats, à l'optimum trouvé se fera comme au premier point, par des visualisations qui compléteront les résultats numériques.

Dans ce travail, nous analyserons les différentes étapes à franchir pour implémenter la stratégie que nous venons de définir.

En particulier, il s'agira de voir quelles implications elle entraîne au niveau algorithmique.

Nous avons fait choix d'algorithmes de type "Newton" ou "quasi newton" dans les développements qui vont suivre; ces méthodes ont en effet largement fait leurs preuves dans les problèmes d'optimisation de fonctions différentiables.

## **Chapitre 2**



## 2. OPTIMISATION D'UNE FONCTION REELLE D'UNE VARIABLE REELLE.

### 2.0. Introduction.

Les algorithmes de minimisation des fonctions d'une variable réelle à valeur réelle ont une grande importance dans le domaine de l'optimisation. Non seulement ils résolvent les problèmes d'optimisation uni-dimensionnels mais en plus, ils prennent une place prépondérante dans les algorithmes multi-dimensionnels.

Dans ce chapitre, nous exposerons deux algorithmes de ce type :

- la méthode de Newton.
- la méthode par interpolation cubique.

Nous avons choisi la méthode de Newton pour des raisons didactiques. Elle est une des méthodes les plus enseignées bien que rarement utilisée en pratique.

Nous avons ajouté au code traduisant l'algorithme deux routines graphiques.

- l'une à caractère local pour permettre de visualiser les itérés successifs de la méthode.
- l'autre pour donner une vue générale de la minimisation.

Ce faisant, nous sommes en mesure de visualiser divers aspects de l'optimisation tels que :

- le choix du point de départ.
- la convergence de la méthode ( vers un maximum ou un minimum ).
- les problèmes de divergences éventuelles.

La méthode par interpolation cubique a été choisie parce qu'elle est souvent utilisée pour effectuer les recherches linéaires dans les algorithmes d'optimisation multi-dimensionnels où les dérivées premières sont requises.

La routine graphique ajoutée au code numérique de cet algorithme peut apporter, au point de vue pédagogique, l'illustration d'un algorithme "hybride" qui est la combinaison d'une méthode comparant des valeurs de fonction et d'un autre utilisant des approximations polynômiales.

Elle permet ainsi de montrer les caractéristiques fondamentales d'une telle approche :

- convergence assurée.
- vitesse de convergence super-linéaire dans certains cas.
- applicable aux problèmes avec contraintes de bornes.



## 2.1. Méthode de Newton.

### 2.1.1. Description de l'algorithme.

Soit à minimiser la fonction réelle  $f$ , trois fois continûment différentiable sur l'intervalle fermé  $[a, b]$ , c'est-à-dire trouver  $x^*$  tel que

$$f(x^*) = \min f(x) \text{ pour } a \leq x \leq b$$

L'idée de la méthode de Newton est d'approximer localement la fonction  $f$  par une fonction quadratique pour laquelle la recherche de l'extremum est immédiate. On construit ainsi une suite d'itérés  $x_1, x_2, \dots, x_r, \dots$  dont on espère qu'elle converge vers la solution  $x^*$  du problème.

La série de Taylor de  $f$  limitée au second ordre au voisinage du point  $x_k$  s'écrit :

$$q(x) = f(x_k) + g(x_k)(x-x_k) + 1/2 (x-x_k)^2 H(x_k)$$

où  $g(x_k)$  est la dérivée première de  $f$  au point  $x_k$  et  $H(x_k)$  est la dérivée seconde de  $f$  au même point

$q(x)$  atteint son extrémum au point

$$x_{k+1} = x_k - \frac{g(x_k)}{H(x_k)} \quad (1)$$

Considérons le développement de Taylor de la dérivée de  $f$ , soit  $g$  autour de  $x_k$

## 2.1. Méthode de Newton.

### 2.1.1. Description de l'algorithme.

Soit à minimiser la fonction réelle  $f$ , trois fois continûment différentiable sur l'intervalle fermé  $[a, b]$ , c'est-à-dire trouver  $x^*$  tel que

$$f(x^*) = \min f(x) \text{ pour } a \leq x \leq b$$

L'idée de la méthode de Newton est d'approximer localement la fonction  $f$  par une fonction quadratique pour laquelle la recherche de l'extérieur est immédiate. On construit ainsi une suite d'itérés  $x_1, x_2, \dots, x_r, \dots$  dont on espère qu'elle converge vers la solution  $x^*$  du problème.

La série de Taylor de  $f$  limitée au second ordre au voisinage du point  $x_k$  s'écrit :

$$q(x) = f(x_k) + g(x_k)(x-x_k) + 1/2 (x-x_k)^2 H(x_k)$$

où  $g(x_k)$  est la dérivée première de  $f$  au point  $x_k$  et  $H(x_k)$  est la dérivée seconde de  $f$  au même point

$q(x)$  atteint son extrémum au point

$$x_{k+1} = x_k - \frac{g(x_k)}{H(x_k)} \quad (1)$$

Considérons le développement de Taylor de la dérivée de  $f$ , soit  $g$  autour de  $x_k$

$$g(x) = g(x_k) + H(x_k)(x - x_k) + \frac{z(\xi_k)}{2} (x - x_k)^2$$

où  $H$  est la dérivée première de  $g$ ,

$z$  est la dérivée seconde de  $g$

$$\text{et } \xi_k \in \begin{cases} (x, x_k) & \text{si } x_k > x \\ (x_k, x) & \text{si } x_k < x \end{cases}$$

A l'extrémum  $x^*$  de la fonction  $f$ , nous avons aussi

$$g(x^*) = 0 = g(x_k) + H(x_k)(x^* - x_k) + \frac{z(\xi_k)}{2} (x^* - x_k)^2$$

et comme

$$x_{k+1} = x_k - \frac{g(x_k)}{H(x_k)}$$

il vient

$$H(x_k)(x^* - x_{k+1}) + 1/2 z(\xi_k)(x^* - x_k)^2 = 0$$

d'où

$$|x^* - x_{k+1}| = 1/2 \frac{|z(\xi_k)|}{|H(\xi_k)|} |x^* - x_k|^2$$

alors si  $|z(x)| \leq M$  et  $|H(x)| \geq N > 0 \quad \forall x \in [a, b]$  (où l'intervalle  $[a, b]$  est supposé contenir tous les itérés de l'algorithme) nous sommes assurés de la convergence quadratique.

$$|x^* - x_{k+1}| \leq \frac{M}{N} |x^* - x_k|^2$$

### Remarques.

La méthode de Newton est une procédure itérative recherchant un point stationnaire de la fonction  $f$  ( $g = 0$ ) lequel n'est pas nécessairement un minimum.

Une divergence peut apparaître si  $H(x_k)$  s'annule ou tend vers zéro (point d'inflexion)

La convergence quadratique est assurée si la fonction est convexe ou concave ( $|H(x)| \geq N > 0$ ) sur  $[a, b]$ . Cet extrémum doit être une racine simple de  $g(x)$

En effet par le théorème des accroissements finis, nous avons que :

$$\begin{aligned} |g(x)| &= |g(x) - g(x^*)| \quad \text{où } x \in [a, b] \\ &= |H(\xi)(x - x^*)| \quad \text{où } \xi \in \begin{cases} (x, x^*) & \text{si } x < x^* \\ (x^*, x) & \text{si } x \geq x^* \end{cases} \\ &\geq N |x - x^*| \end{aligned}$$

d'où

$$|g(x)| = 0 \Leftrightarrow x = x^*$$

de plus si l'extrémum n'est pas simple alors

$$|x^* - x_{k+1}| \leq \frac{M}{N} |x^* - x_k|^2$$

Remarques.

La méthode de Newton est une procédure itérative recherchant un point stationnaire de la fonction  $f$  ( $g = 0$ ) lequel n'est pas nécessairement un minimum.

Une divergence peut apparaître si  $H(x_k)$  s'annule ou tend vers zéro (point d'inflexion)

La convergence quadratique est assurée si la fonction est convexe ou concave ( $|H(x)| \geq N > 0$ ) sur  $[a, b]$ . Cet extrémum doit être une racine simple de  $g(x)$

En effet par le théorème des accroissements finis, nous avons que :

$$|g(x)| = |g(x) - g(x^*)| \quad \text{où } x \in [a, b]$$

$$= |H(\xi)(x - x^*)| \quad \text{où } \xi \in \begin{cases} (x, x^*) & \text{si } x < x^* \\ (x^*, x) & \text{si } x \geq x^* \end{cases}$$

$$\geq N |x - x^*|$$

d'où

$$|g(x)| = 0 \Leftrightarrow x = x^*$$

de plus si l'extrémum n'est pas simple alors

$$g(x) = (x-x^*)^k q(x) \quad \text{où } q(x^*) \neq 0 \text{ et } k > 1$$

d'où

$$H(x) = (x-x^*)^{k-1} q(x) + (x-x^*)^k q'(x)$$

$$= (x-x^*)^{k-1} w(x)$$

$$\text{où } W(x) = q(x) + (x-x^*)q'(x)$$

par conséquent

$$|H(x)| = |(x-x^*)^{k-1}| |W(x)| \geq N > 0$$

or si  $x \rightarrow x^*$  alors  $|H(x)| \rightarrow 0$

Ce qui prouve que la convergence quadratique n'est assurée que si l'extrémum est une racine simple de  $g(x)$

Lorsque  $g(x)$  admet une racine multiple la convergence peut devenir plus lente que celle de la méthode dichotomique.



### 2.1.2. Implémentation du graphique.

Les informations fournies à chaque itération de la méthode de Newton sont :

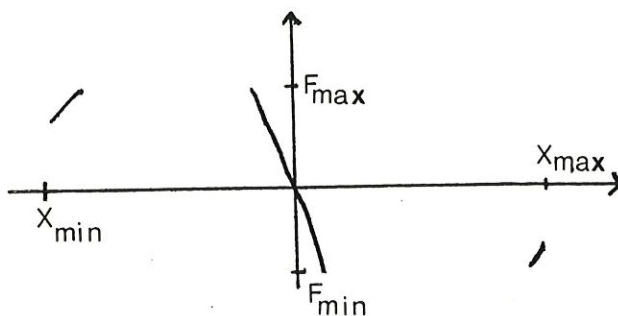
- $x$  : l'avant dernière estimation de l'extrémum
- $y$  : l'estimation de l'extrémum.
- $G$  : la valeur de la dérivée première de la fonction au point  $x$
- $H$  : la valeur de la dérivée seconde de la fonction au point  $x$ .

En utilisant  $x, G, H$  et l'évaluation de la fonction au point  $x$ , nous pouvons reconstruire l'approximation quadratique de la fonction, lesquelles peuvent alors être représentées graphiquement.

Le problème crucial de la représentation graphique d'une fonction d'une variable réelle est le choix des échelles sur chacun des axes  $x$  et  $y$  ( $= f(x)$ ). Parmi les inconvénients qu'on peut rencontrer citons les suivants :

- si l'échelle en  $y$  est trop grande, il peut en résulter un "aplatissement" de la fonction.
- ou encore une représentation telle que décrite sur la figure ci-dessous risque de se produire si l'échelle en  $Y$  est trop petite.

Fig (0)



D'où l'importance du choix des échelles graphiques

Nous avons choisi de fournir en cours d'algorithme à l'utilisateur une échelle des x et des y, par défaut, qui se sont révélées être de bons choix.

L'échelle en x est définie par les expressions :

$$x_{\min} = \begin{cases} x - (y-x)/2 & \text{si } x < y \\ y - (x-y)/2 & \text{si } x > y \end{cases}$$

$$x_{\max} = \begin{cases} y + (y-x)/2 & \text{si } x < y \\ x + (x-y)/2 & \text{si } x > y \end{cases}$$

et celle en y par :

$$y_{\min} = \min_z F(z)$$

$$y_{\max} = \max_z F(z)$$

$$\text{où } z \in \left\{ V \mid V = x_{\min} + h \cdot p \quad \text{où } h \in \{0, 1, \dots, 10\} \text{ et } p = \frac{|x-y|}{5} \right\}$$



Nous tracerons en plus le segment de droite allant du point  $(X, F(x))$  au point  $(Y, F(y))$  pour faire ressortir le changement de la valeur de la fonction.

Lorsque l'algorithme numérique est terminé, une vue générale des différents itérés est présentée. Le domaine de visualisation de la fonction est l'intervalle  $(a, b]$  de départ et l'intervalle  $(y_{\min}, y_{\max}]$  sur l'axe  $y$  est déterminé de manière analogue à celui des vues successives

$$Y_{\min} = \min_z F(z)$$

$$y_{\max} = \max_z F(z)$$

$$\text{où } z \in \left\{ V \mid V = \bar{a} + k \times P \quad \text{avec } k \in \{0, 1, \dots, 20\} \text{ et } p = (b - \bar{a}) / 20 \right\}$$

L'algorithme de tracé des différentes approximations s'écrit :

# 1       $k = 1$

# 2      si  $(k < k_{\max} \text{ ou } |x^{(k)} - x^{(k+1)}| > (B-A) / 10)$

alors aller en #3

sinon STOP.

# 3 : Définition de l'intervalle de représentation de la  $k^{\text{ième}}$  approximation sur l'axe X

$$x_{\min}^{(k)} = \begin{cases} x^{(k)} - (x^{(k+1)} - x^{(k)})/2, & \text{si } x^{(k)} < x^{(k+1)} \\ x^{(k+1)} - (x^{(k)} - x^{(k+1)})/2, & \text{si } x^{(k)} > x^{(k+1)} \end{cases}$$

$$x_{\max}^{(k)} = \begin{cases} x^{(k+1)} + (x^{(k+1)} - x^{(k)})/2, & \text{si } x^{(k)} < x^{(k+1)} \\ x^{(k)} + (x^{(k)} - x^{(k+1)})/2, & \text{si } x^{(k)} > x^{(k+1)} \end{cases}$$

# 4 : tracé de la  $k^{\text{ième}}$  approximation quadratique.

# 5 : Tracé du segment défini par les points  $(x^{(k)}, f(x^{(k)}))$  et  $(x^{(k+1)}, f(x^{(k+1)}))$

# 6 :  $k = k + 1$  et retour en 2.

#### Remarques.

. KMAX est le nombre d'itérations effectuées par l'algorithme

. La condition  $|x^{(k)} - x^{(k+1)}| > (B-A) / 10$  assure une plus grande clarté du graphique dans la région de l'optimum, car lorsque nous approchons de l'extrémum, les approximations quadratiques successives se rapprochent de plus en plus et le résultat des différents tracés se résume à une grosse tache.

• Les formules du pas  $\frac{1}{3}$  ont également été choisies pour parvenir à un graphique lisible.

### 2.1.3. Exemples.

#### 2.1.3.1. Premier cas d'application.

Nous avons fait le choix de la fonction.

$$f(x) = \sin(x) + \frac{1}{2} \sin(2x) + \frac{1}{3} \sin(3x)$$

à minimiser sur l'intervalle  $(0,6)$

Soit  $x^0=3$ , la figure 1 illustre le comportement de l'algorithme de Newton lors de la première itération. Nous nous apercevons que le point de départ est mal choisi.

La figure 2 montre qu'à l'itération suivante on converge vers un minimum local.

La figure 3 donne une vue générale de la minimisation. Elle nous permet de mesurer l'écart entre la solution obtenue et le minimum global.

### 2.1.3. Exemples.

#### 2.1.3.1. Premier cas d'application.

Nous avons fait le choix de la fonction.

$$f(x) = \sin(x) + \frac{1}{2} \sin(2x) + \frac{1}{3} \sin(3x)$$

à minimiser sur l'intervalle  $(0,6)$

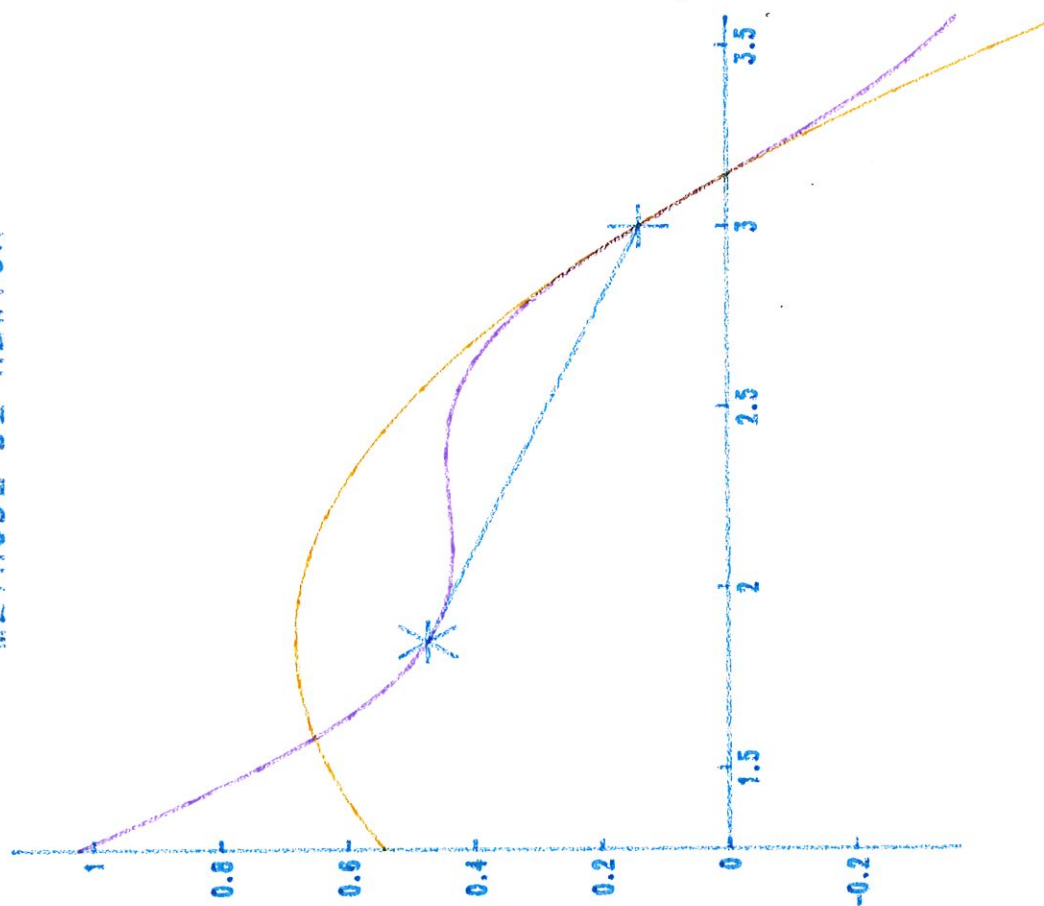
Soit  $x^0=3$ , la figure 1 illustre le comportement de l'algorithme de Newton lors de la première itération. Nous nous apercevons que le point de départ est mal choisi.

La figure 2 montre qu'à l'itération suivante on converge vers un minimum local.

La figure 3 donne une vue générale de la minimisation. Elle nous permet de mesurer l'écart entre la solution obtenue et le minimum global.

# METHODE DE NEWTON

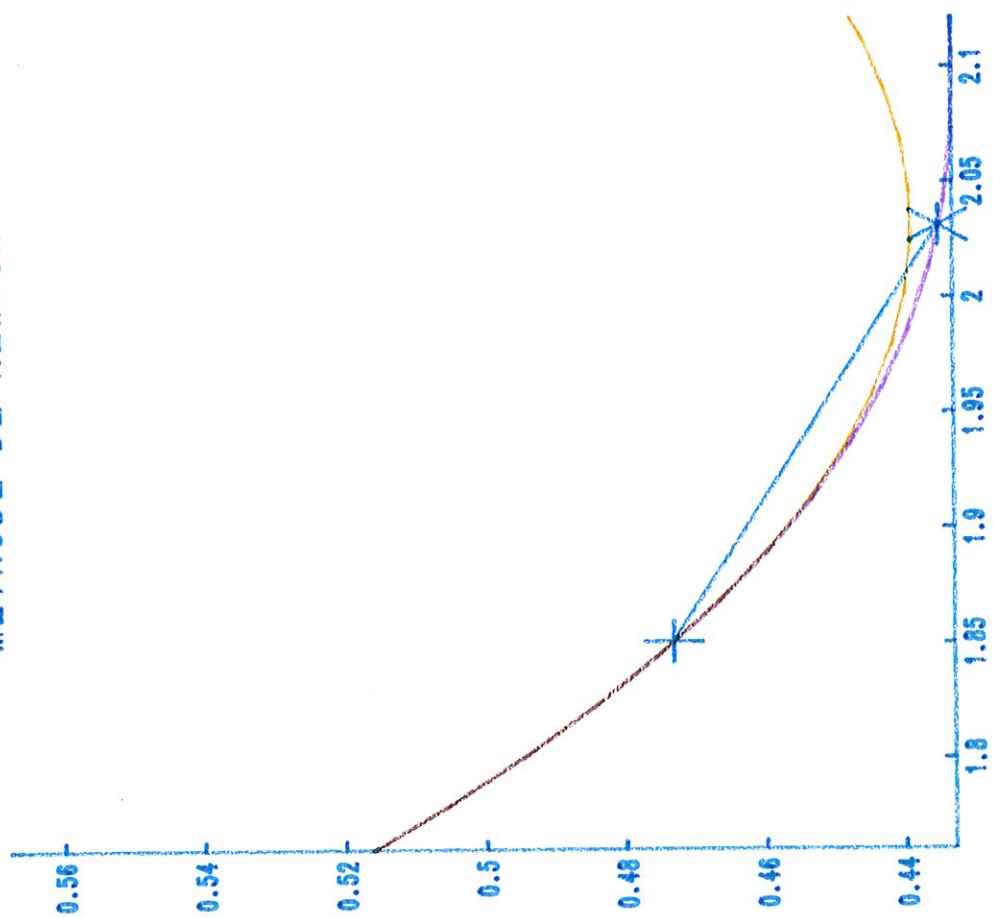
FIGURE 1



ITERATION 0  
 $X = 0.300000E+01$   
 $FX = 0.13879E+00$   
 $Y = 0.18506E+01$   
 $FY = 0.47305E+00$

# METHODE DE NEWTON

FIGURE 2



ITERATION 1  
 $X = 0.18506E+01$   
 $FX = 0.47305E+00$   
 $Y = 0.20309E+01$   
 $FY = 0.43503E+00$

METHOD OF NEWTON

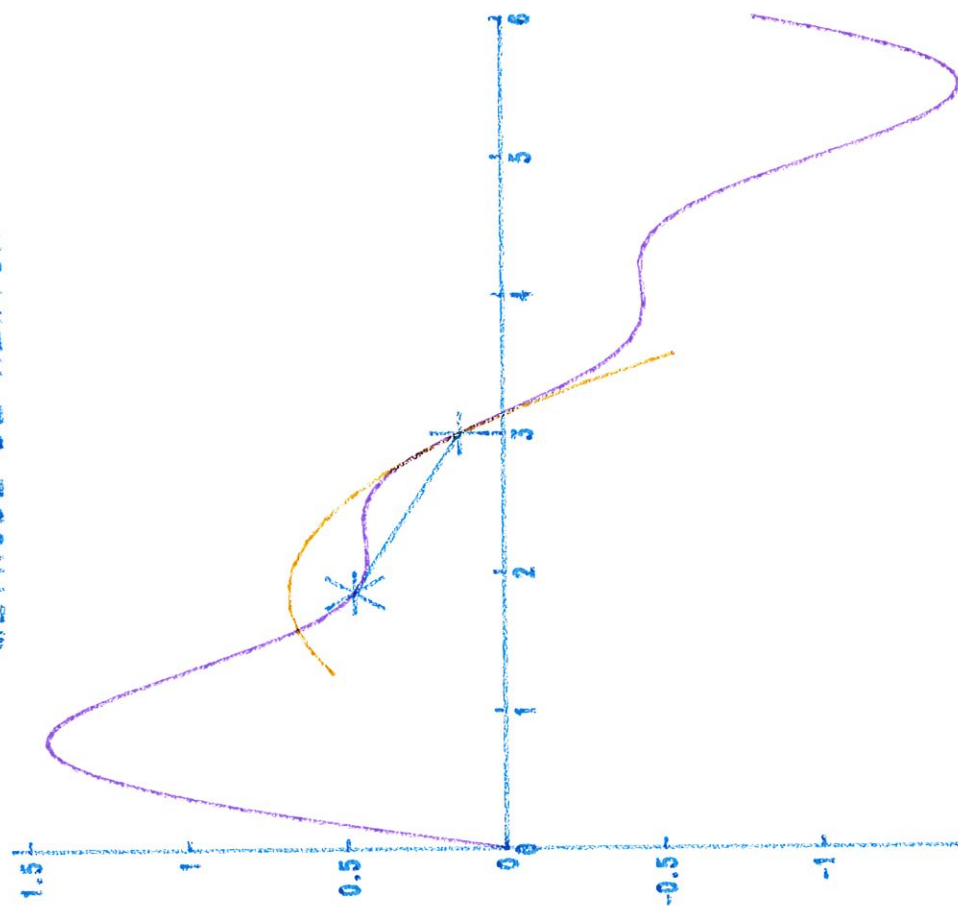


FIGURE 3



### 2.1.3.2. Deuxième cas d'application.

Soit la fonction,  $\frac{x^5}{5} - \frac{5}{3} x^3 + 4x + 7$  à minimiser sur l'intervalle  $(-1.5; 3.5)$ .

Soit  $x^0 = 3$ , la figure 4 donne la vue générale d'une minimisation sur l'intervalle réduit  $(-0.5, 3.5)$ . Nous constatons la convergence de l'algorithme vers un minimum local égale à 2.

Le minimum global de la fonction sur l'intervalle réduit se situant en  $x = -0.5$  ( c'est-à-dire à la borne inférieure de l'intervalle), nous avons alors relancé la minimisation en ce point mais cette fois, sur l'intervalle  $(-1.5, 2)$  ; la figure 5 montre la convergence globale vers le point  $x = -1$ .

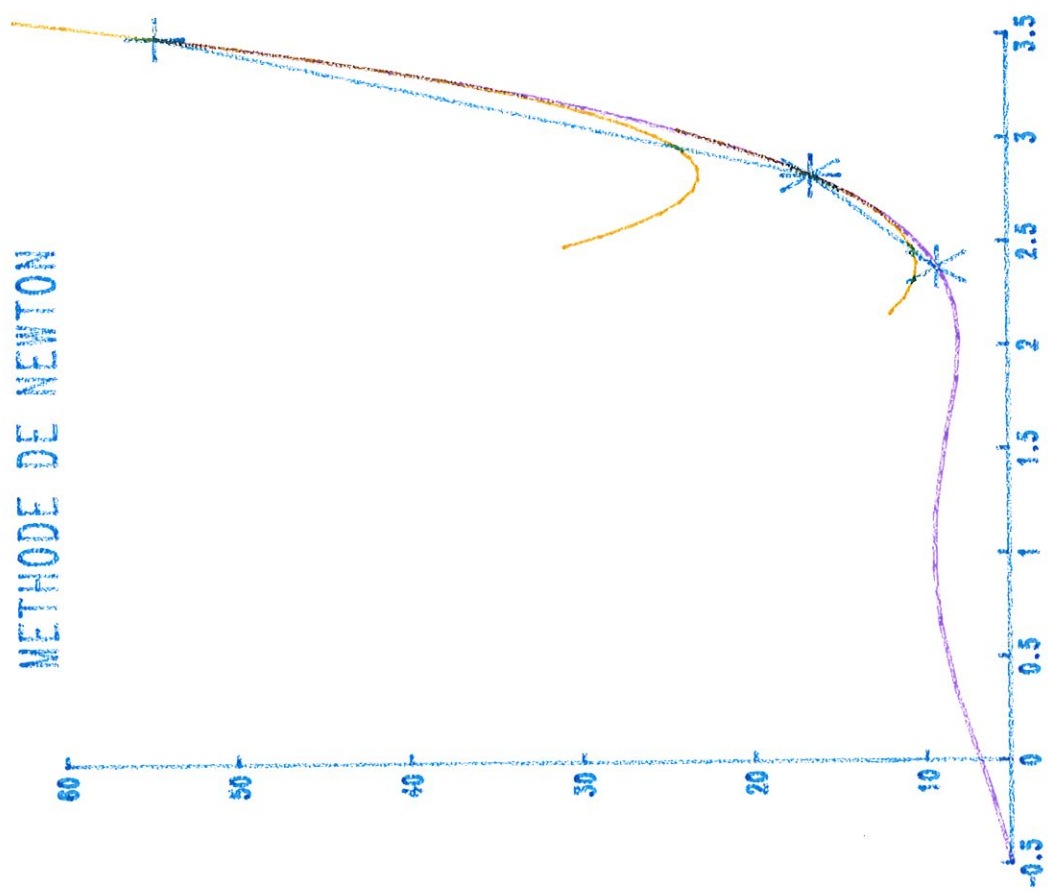
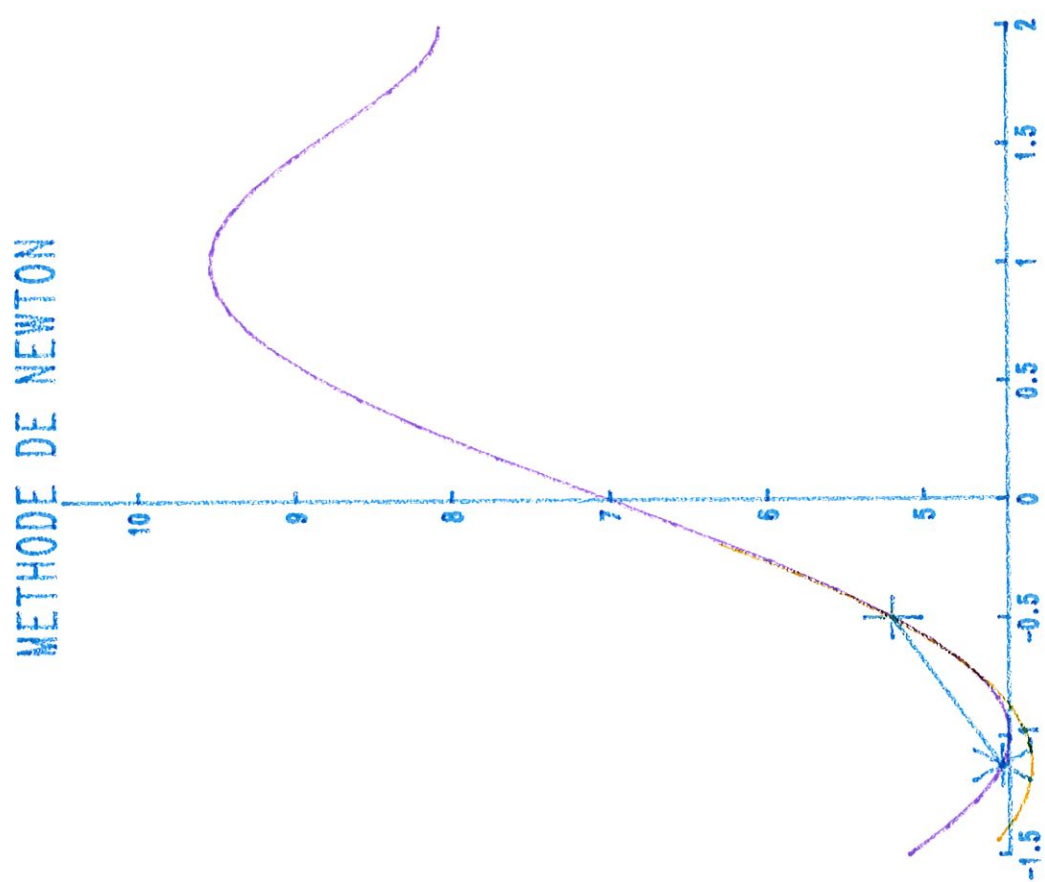


FIGURE 4

FIGURE 5



## 2.2. Méthode par interpolation cubique.

Soit à minimiser la fonction  $f$ , deux fois continûment différentiable, sur l'intervalle  $[a, b]$

Soient  $g(x)$  et  $G(x)$  respectivement les dérivées première et seconde de  $f(x)$ .

Dans la méthode de Newton, la recherche de l'extrémum s'effectue au moyen d'approximations quadratiques :

$$q(h) = f(y) + h g(y) + \frac{1}{2} h^2 G(y)$$

En effet, soit  $y$ , un des itérés successifs de la méthode Newton, il vient :

$$q(0) = f(y) \quad (1)$$

$$q'(0) = g(y) \quad (2)$$

$$q''(0) = G(y) \quad (3)$$

Si la dérivée seconde n'est pas connue l'équation (3) peut être remplacée par une autre du type

$$q(w) = f(w)$$

$w$  étant par exemple l'itéré précédent. Dans ce cas nous disposons aussi de la dérivée  $g(w)$  d'où l'idée d'approximer la fonction par un polynôme cubique lequel possède également un minimum facilement calculable.

L'équation du polynôme cubique  $P$  passant par les points  $(x, f(x)), (w, f(w))$  avec des dérivées aux points  $x$  et  $w$  respectivement égales à  $g(x)$  et  $g(w)$  est donné par :

$$P(z) = f(x) + g(x)z - \frac{z^2}{(w-x)} (g(x) + \eta) + \frac{z^3}{3(w-x)^2} (g(x) + g(w) + 2\eta)$$

où

$$\eta = 3 \frac{f(x) - f(w)}{(w-x)} + g(x) + g(w)$$

La dérivée du polynôme P vaut :

$$P'(z) = g(x) - \frac{2z}{(w-x)} (g(x) + \eta) + \frac{z^2}{(w-x)} (g(x) + g(w) + 2\eta)$$

La racine de  $P'(z) = 0$  correspondant à un minimum de l'interpolation cubique est donnée par l'expression  $X + S/q$

$$\text{où } S = \pm (w-x) (g(x) - \eta - \gamma)$$

$$q = \mp (g(w) - g(x) + 2\gamma) \quad (4)$$

$$\text{et } \gamma = \text{signe } (w-x) (\eta^2 - g(x)g(w))^{\frac{1}{2}}$$

Cette formule a été suggérée par Davidon (1959).

Si l'intervalle  $(x, w)$  contient le minimum  $x^*$  de la fonction  $f$  alors celui du polynôme d'interpolation  $P$  appartient aussi à l'intervalle (sauf erreurs d'arrondi).

Un avantage de cette méthode est que sa vitesse de convergence est super linéaire, c'est-à-dire que

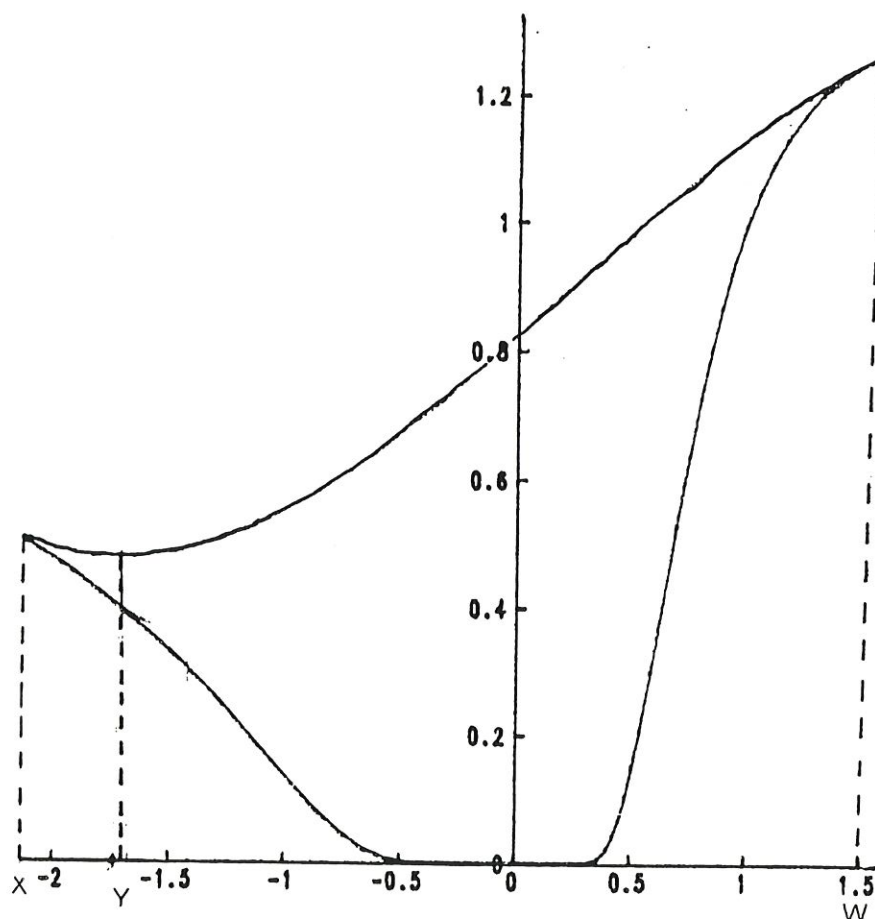
$$\lim_{k \rightarrow \infty} \frac{|x^{(k+1)} - x^*|}{|x^{(k)} - x^*|} = 0,$$

si le point de départ est suffisamment proche de la solution.

Cette méthode basée sur l'utilisation des formules (4) implique qu'on écarte un point à chaque étape.

Si, comme dans le cas de méthodes comparant des valeurs de fonction (dichotomie, section d'or,...), le point écarté est tel que le nouvel ensemble de point renferme encore le minimum, une grande valeur de la fonction peut être retenue durant plusieurs itérations et la vitesse de convergence risque d'en souffrir ( Fig (1) ).

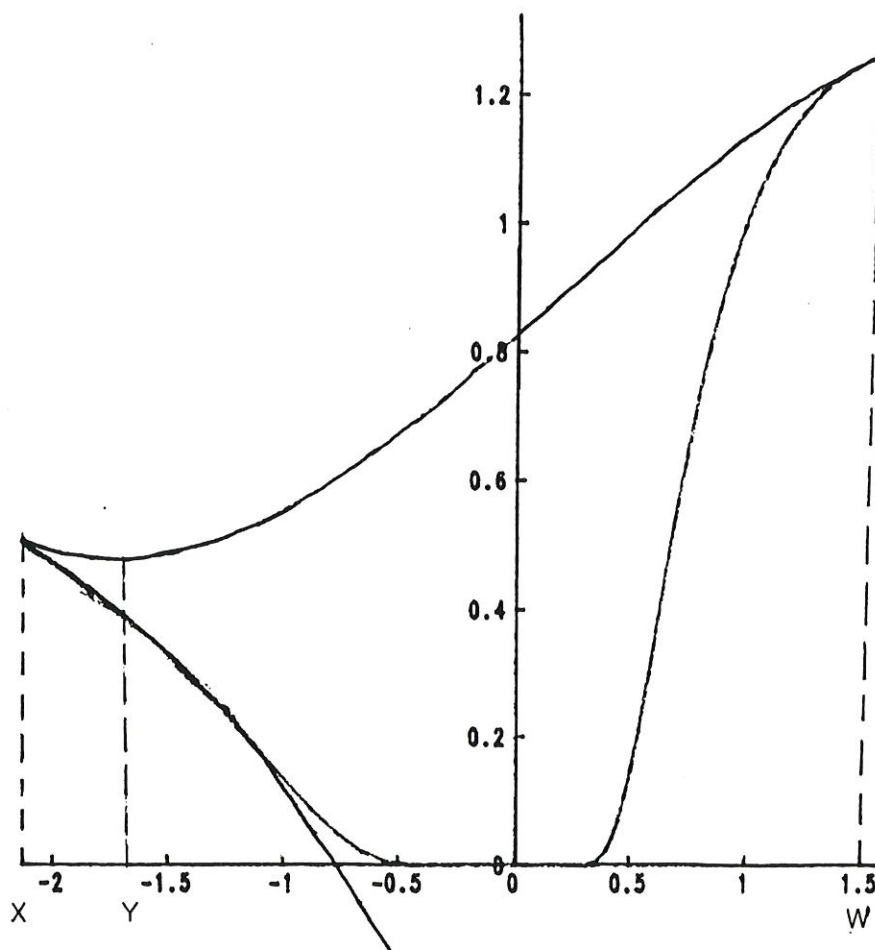
(Fig 1)



La figure 1 illustre le problème. Le minimum de l'interpolation cubique passant par  $x$  et  $w$  sera très proche de  $x$  et les itérés suivants s'en éloigneront très lentement.

Une alternative naturelle pour résoudre ce problème est d'écarter le point correspondant à la plus grande valeur de la fonction, mais dans ce cas, malheureusement, l'intervalle défini par le nouvel ensemble de points peut ne plus contenir le minimum et la suite des itérés risque de diverger ( Fig 2 ).

Fig(2)



La figure 2 décrit l'alternative présentée ci-dessus. Le minimum  $y$  de l'interpolation passant par  $x$  et  $w$  sera très proche de  $x$ , et celui de l'interpolation passant par  $x$  et  $y$  sortira de l'intervalle  $[x, w]$ .



Les désavantages des méthodes comparant des valeurs de fonction et d'approximation polynomiale peuvent être éliminés en les combinant.

Cette classe d'algorithmes à l'avantage de posséder la convergence des méthodes comparant les valeurs de fonction ainsi qu'une vitesse de convergence super-linéaire ( asymptotiquement ) grâce aux approximations polynomiales successives.

Les concepts de base de tels algorithmes sont :

- de rechercher un intervalle  $[x, w]$  renfermant le minimum, c'est-à-dire un intervalle tel que le théorème des valeurs intermédiaires garantisse l'existence d'une racine de  $g$  dans  $[x, w]$ .
- de toujours calculer les approximations polynomiales successives à l'aide des deux meilleurs points obtenus.
- de diminuer la longueur de l'intervalle renfermant le minimum.
- d'utiliser les approximations polynomiales jusqu'au moment où le point prédit sort de l'intervalle renfermant le minimum. Dans ce cas, on choisit l'itéré égal à un point calculé qui porte le nom de borne d'extrapolation (application de la méthode comparant des valeurs de fonction ).

La longueur de l'intervalle  $[x, w]$  renfermant le minimum sera diminuée de la manière suivante :

Soit  $y$  l'itéré obtenu.

- si  $g(y) > 0$  alors le nouvel intervalle sera  $[x, y]$  car le théorème des valeurs intermédiaires nous garantit l'existence d'un minimum de  $f$  dans cet intervalle.
- si  $g(y) < 0$  alors le nouvel intervalle sera  $[y, w]$ .

La borne d'interpolation  $M$ , dans le cas où un intervalle  $(c, d)$  contient le minimum connu, peut être choisie de la façon suivante :

$$M = \begin{cases} x + \frac{1}{2} (-I_1 \ I_2)^{\frac{1}{2}} & \text{si } |I_1| < |I_2| \quad (5) \\ x + \frac{5}{11} (0,1 - I_2 / I_1) I_2 & \text{si } |I_1| \geq |I_2| \quad (6) \end{cases}$$

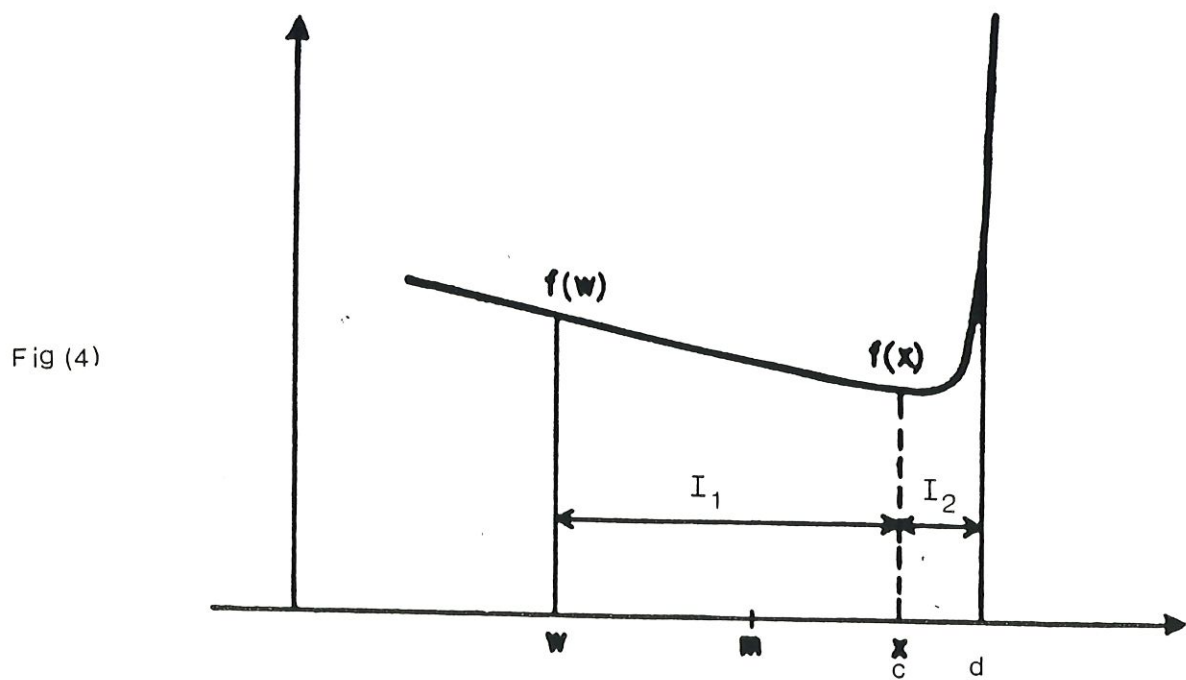
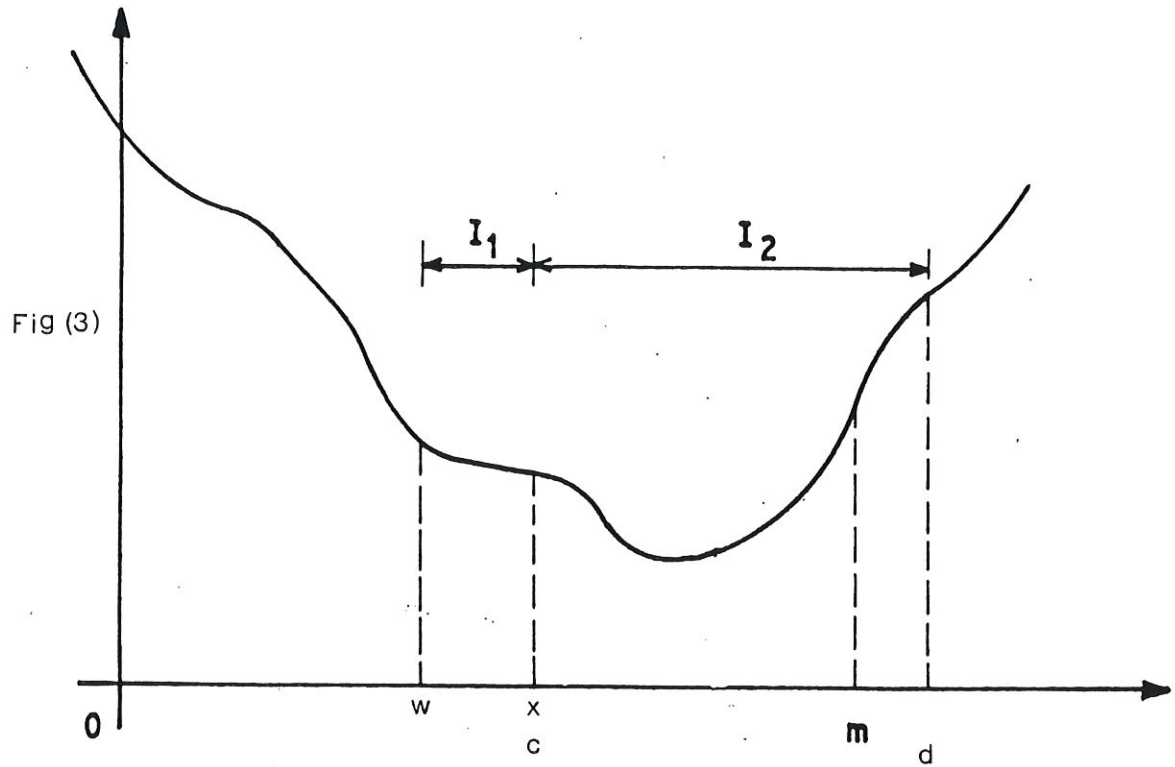
$$\text{où } I_1 = x - w \text{ et } I_2 = \begin{cases} d - x & \text{si } w < x \\ c - x & \text{si } w > x \end{cases}$$

avec  $x$  et  $w$  les deux meilleurs points obtenus tel que  $f(x) < f(w)$ .

Cette borne d'extrapolation a été proposée par Gill et Murray (1972).

Remarquons que si  $|I_1| = |I_2|$  la formule fournit le point milieu de  $[x, d]$  si  $w < x$  et celui de  $[c, x]$  si  $x < w$  et que  $I_1$  et  $I_2$  sont toujours de signes opposés (Fig 3 et Fig 4)

Les figures 3 et 4 illustrent respectivement les formules 4 et 5



Dans ce qui précède nous avons supposé qu'un intervalle d'incertitude était connu. Dans le cas contraire la borne d'extrapolation  $M$  peut être fournie par l'équation

$$M = x + 4 (x - w) \quad (7)$$

où  $x$  et  $w$  sont les deux meilleurs points connus et  $f(x) < f(w)$ .

L'avantage d'une telle borne est la croissance rapide de  $M$  lorsque l'extrapolation se répète.

#### Critère d'arrêt.

Nous considérons que l'algorithme a convergé si le minimum est compris dans un intervalle  $[c, d]$  telle que

$$d - c \leq 2 \text{ tol}(y) \quad (8)$$

où  $x$  est la meilleure estimation connue du minimum et  $\text{tol}(x) = \varepsilon |X| + \tau$  avec  $\varepsilon$  et  $\tau$  respectivement la précision relative et absolue sur le minimum.

#### Restriction sur le pas.

Le fait de minimiser une fonction réelle à valeur réelle sur un intervalle  $[A, B]$  nous oblige à imposer une borne supérieure et inférieure sur le point prédit pour que les itérés successifs restent admissibles.

Nous allons traiter uniquement le cas d'une borne supérieure. Un raisonnement analogue à celui qui suit peut s'appliquer pour une borne inférieure.

Soit  $\lambda = B$ , nous définissons :

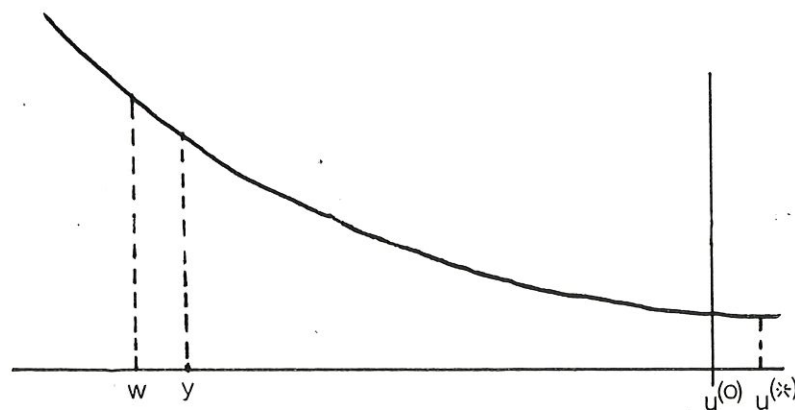
- une borne supérieure artificielle  $u^{(*)} = \lambda + \text{tol}(\lambda)$
- une borne supérieure temporaire  $u^{(0)} = \lambda$

La fonction ne sera jamais évaluée en  $u^{(*)}$ . La raison du choix de ces deux bornes apparaîtra plus tard.

Supposons qu'après  $K$  itérations

- $y$  et  $w$  sont les deux meilleurs points.
  - aucun intervalle renfermant le minimum n'est connu.
  - le point résultant de l'interpolation cubique passant par  $y$  et  $w$  dépasse la borne d'extrapolation
- $$M = y + 4 (y - w)$$
- $M > u^{(0)}$

Fig(5)



La figure illustre le cas présenté ci-dessus.

Alors  $u^{(0)}$  est considérée comme la nouvelle estimation du minimum et nous déplaçons la borne temporaire vers la gauche par la formule suivante :

$$u^{(1)} = u^{(0)} - \text{tol}(u^{(0)}) / (1 + \epsilon)$$

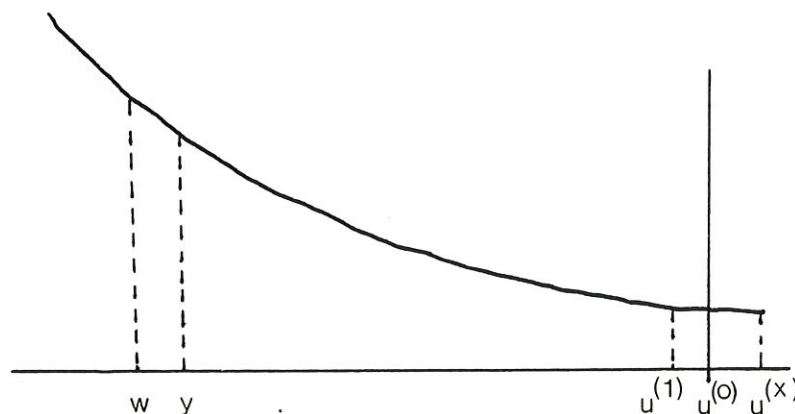
Si comme dans le cas de la Fig 6 , les deux meilleurs points sont  $y$  et  $u^{(0)}$  et que le point obtenu par l'interpolation cubique se trouve au delà de  $u^{(1)}$  alors nous considérons  $u^{(1)}$  comme la nouvelle estimation du minimum.

Et finalement si  $u^{(1)}$  et  $u^{(0)}$  sont les deux meilleurs points ( cas de la Fig . 6 ), nous avons que

$$\left| u^{(0)} - \left( \text{tol}(u^{(0)}) / (1 + \epsilon) \right) - u^{(0)} - \text{tol}(u^{(0)}) \right| < 2 \text{ tol}(u^{(0)})$$

et par conséquent, nous avons atteint l'optimum.

Fig (6)





. Du point de vue algorithmique.

Pour diminuer les erreurs d'arrondi, les différents points seront toujours considérés par rapport au meilleur point connu, soit  $x$ , c'est-à-dire que les quantités  $\bar{c} = c - x$ ,  $\bar{d} = d - x$ ,  $\bar{w} = w - x$ , seront utilisées au lieu de  $c, d, w$ .

Les erreurs d'arrondi dans le calcul peuvent introduire une non-unimodalité. Ce type de problème peut être évité en introduisant la classe des fonctions  $\delta$ -unimodales suggérée par Brent (1973). Une fonction est  $\delta$ -unimodale si elle est unimodale quand on l'évalue en des points qui sont distants d'au moins  $\delta$ , où  $\delta$  est un petit scalaire positif. Par conséquent, dans l'algorithme, il est requis que la nouvelle estimation  $y$  du minimum est telle que les distances  $|y - x|$ ,  $y - c$ ,  $d - y$  sont au moins égales à  $\text{tol}(x)$  où  $\text{tol}(x) = \varepsilon |x| + \tau$ . Si  $y$  ne satisfait pas ces critères un pas de longueur  $\text{tol}(x)$  est pris.

Suivant l'itéré de Brent (1973), l'algorithme utilise un pas d'une méthode comparant des valeurs de fonction si

- $q = 0$
  - $x + s/q$  dépasse une borne d'extrapolation.
  - si  $|s/q| \geq \frac{1}{2} |e|$
  - $|e| \leq \text{tol}(x)$
- } (9)

où  $e$  est le pas pris à l'avant dernière itération.

et  $s/q$  est le pas obtenu par l'équation (4).



$\left| \frac{s}{q} \right| < \left| \frac{1}{2} e \right|$  assure que le pas diminue d'au moins d'un facteur deux toutes les deux itérations.

En résumé, l'algorithme prendra en autre :

- un pas d'extrapolation si les deux meilleurs points définissent l'intervalle renfermant le minimum et si une des conditions (9) est vérifiée.

- la nouvelle estimation du minimum égale à la borne d'extrapolation donnée par les équations (5) et (6) si un intervalle renfermant le minimum est connu mais que les deux meilleurs points n'encadrent pas le minimum et si une des conditions (9) est vérifiée
- la nouvelle estimation du minimum égale à la borne d'extrapolation de l'équation (7) si aucun intervalle renfermant le minimum n'est connu et si une des conditions (9) est vérifiée.

### 2.2.2. Implémentation graphique.

Les informations fournies par l'algorithme numérique à chaque itération sont les suivantes :

- les points  $(y, F(y))$  et  $(y, F(y))$  engendrant l'interpolation cubique dont les dérivées sont respectivement  $g(x)$  et  $g(y)$
- $U$  la nouvelle estimation du minimum.
- l'intervalle  $(BI, BS)$  renfermant le minimum.

Par conséquent, nous avons la possibilité de reconstruire l'interpolation cubique et de la tracer.

Pour des raisons identiques à celle évoquées dans la section 2.1.2., nous fournissons par défaut à l'utilisateur un intervalle sur l'axe  $x$  et sur l'axe  $y=F(x)$  Ils sont définis de la manière suivante :

- sur l'axe  $x$  :

$$x_{\min} = \begin{cases} y \times 1,05, & \text{si } y < BI \wedge y < 0 \\ y \times 0,95, & \text{si } y < BI \wedge y > 0 \\ BI \times 1,05, & \text{si } y > BI \wedge BI < 0 \\ BI \times 0,95, & \text{si } y > BI \wedge BI > 0 \end{cases}$$

$$x_{\max} = \begin{cases} y \times 0,95, & \text{si } y > BS \wedge y < 0 \\ y \times 1,05, & \text{si } y > BS \wedge y > 0 \\ BS \times 0,95, & \text{si } y < BS \wedge BS < 0 \\ BS \times 1,05, & \text{si } y < BS \wedge BS > 0 \end{cases}$$

- sur l'axe y :

$$F_{\min} = \min_z \left\{ F(z) \right\}$$

$$F_{\max} = \max_z \left\{ F(z) \right\}$$

$$\text{où } z \in \left\{ V \mid V = x_{\min} + h \times \text{pas où } h \in \{0, 1, \dots, 10\} \right. \\ \left. \text{et pas} = (X_{\max} - X_{\min})/10 \right\}$$

Remarques.

Le choix de l'intervalle sur l'axe x permet de visualiser la fonction et l'interpolation cubique sur l'intervalle renfermant le minimum. Si le deuxième point par lequel passe l'interpolation n'appartient pas à l'intervalle renfermant le minimum, l'intervalle sur l'axe x est agrandi de manière à ce que le point soit visualisé.

Nous rappelons que l'estimation du minimum U ne correspond pas nécessairement au minimum de la cubique.

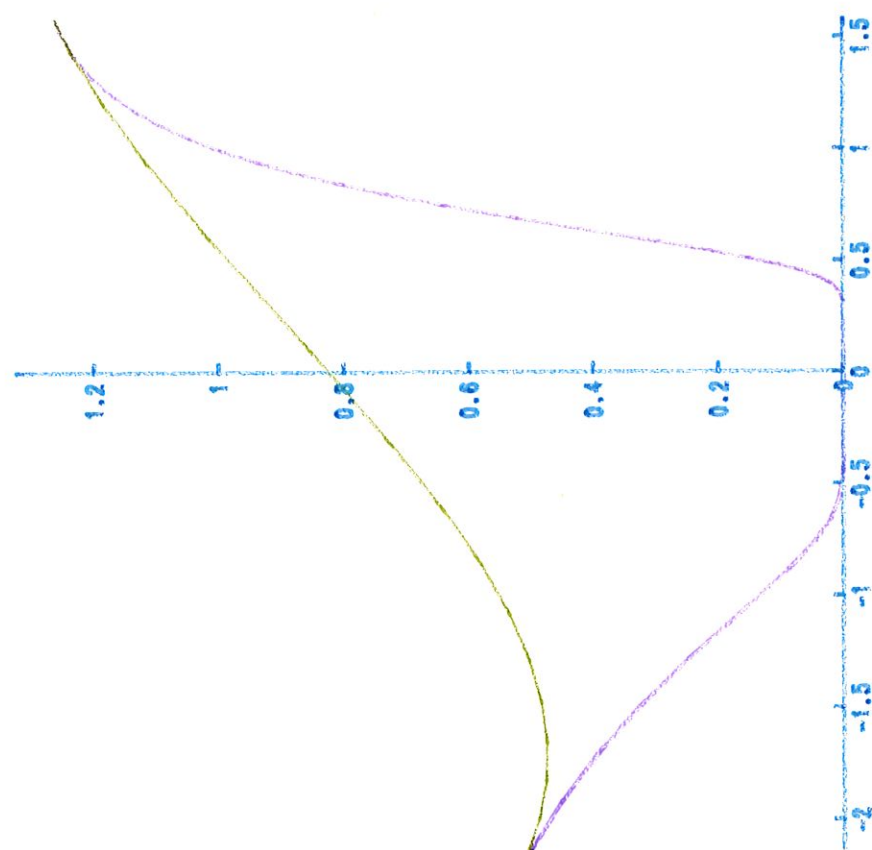
2.2.3. Exemple.

Soit la fonction  $e^{\left(\frac{x-1}{x^2}\right)}$  à minimiser sur l'intervalle  $(-6,9)$ .

Dès la première itération on s'aperçoit du mauvais conditionnement du problème ( Figure 1). La procédure convergera mais très lentement, pour l'accélérer, on relancera la minimisation dans un petit voisinage de zéro. La figure 2,3 et 4 illustrent la convergence lente du problème initial ainsi que la stratégie du pas de calcul.

# METHODE PAR INTERPOLATION CUBIQUE

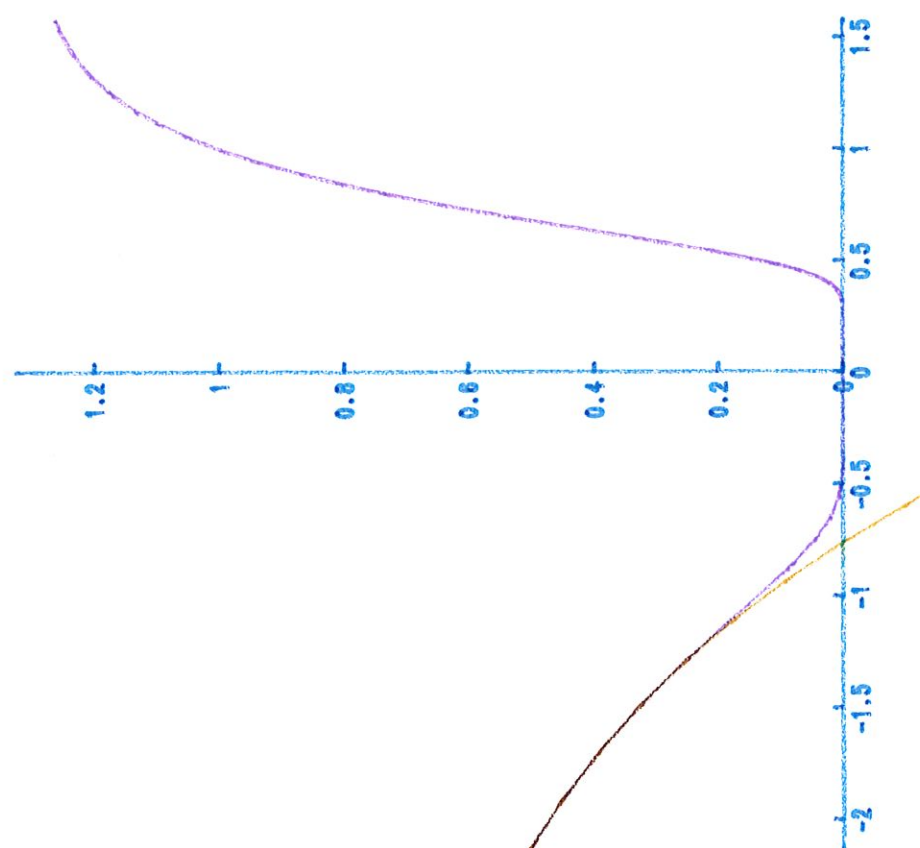
FIGURE 1



ITERATION 1  
X = -0.22500E+01  
FX= 0.52625E+00  
Y = 0.15000E+01  
FY= 0.12488E+01  
U = -0.17260E+01  
R FONCTION  
V AP.CUB.

# METHODE PAR INTERPOLATION CUBIQUE

FIGURE 2

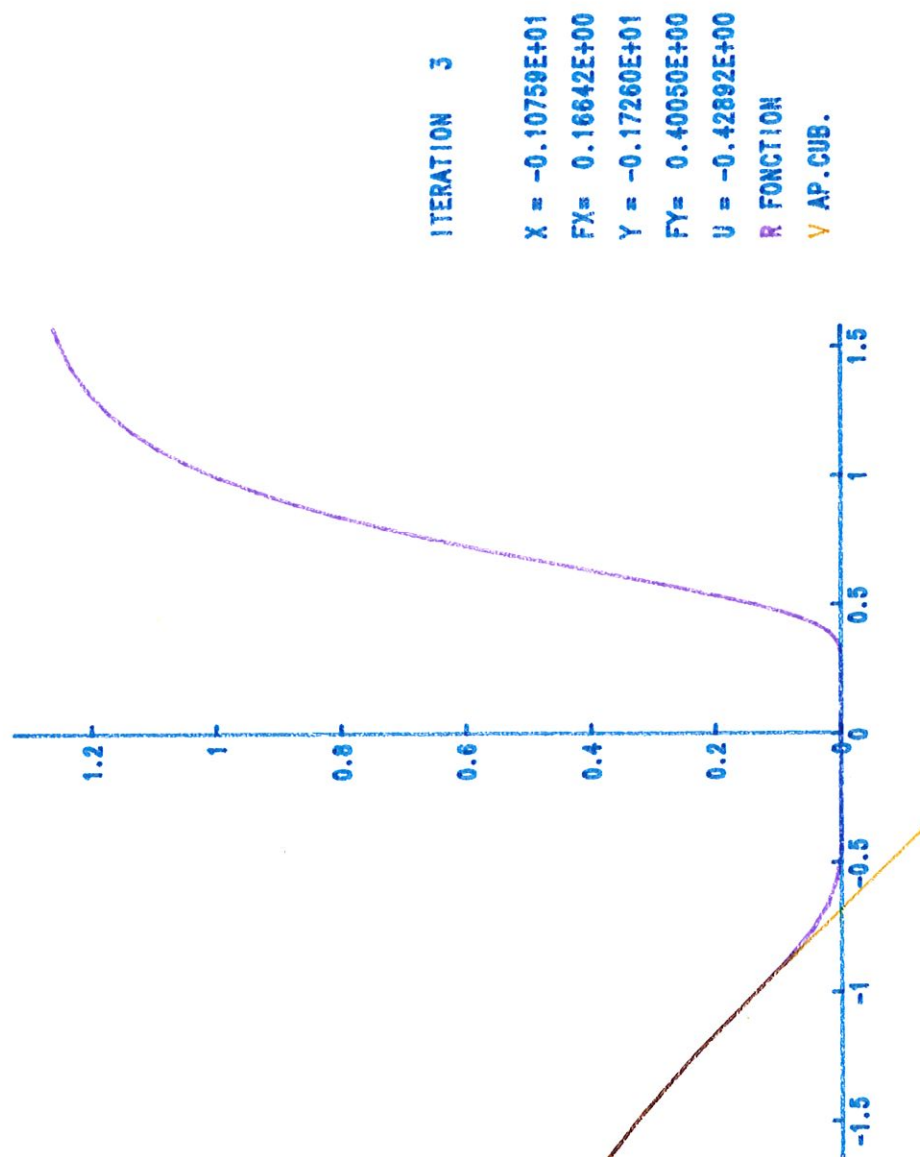


ITERATION 2

X = -0.17260E+01  
FX= 0.40050E+00  
Y = -0.22500E+01  
FY= 0.52625E+00  
U = -0.10759E+01  
R FONCTION  
V AP.CUB.

# METHODE PAR INTERPOLATION CUBIQUE

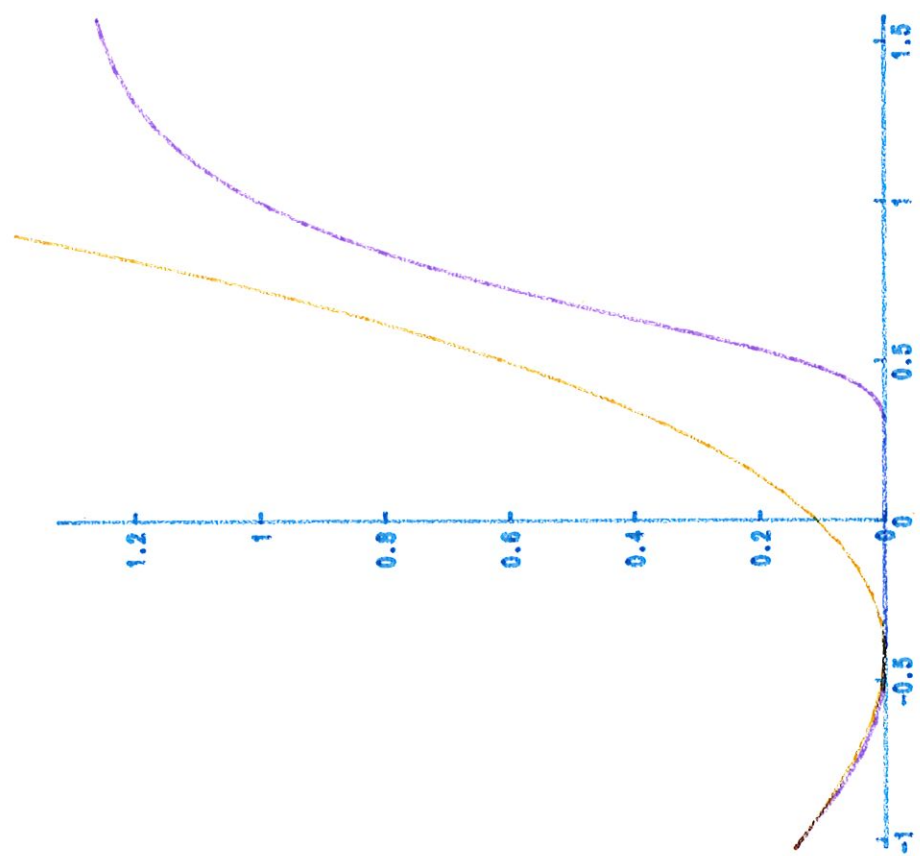
FIGURE 3





# METHODE PAR INTERPOLATION CUBIQUE

FIGURE 4



ITERATION 4

X = -0.42892E+00

FX = 0.42347E-03

Y = -0.10759E+01

FY = 0.16642E+00

U = -0.41634E+00

R FUNCTION

V AP.CUB.

### 2.3. Conclusions.

Les tests effectués montrent que l'utilisation de l'outil graphique peut être considéré comme une aide efficace aux problèmes d'optimisation uni-dimensionnels.

Il permet, entre autre, de pallier aux inconvénients inhérents à ces algorithmes, c'est-à-dire :

- trouver un bon point de départ.
- éviter les problèmes éventuels de divergences ou de convergence lente.
- de localiser aisément un minimum global de la fonction.

L'outil graphique alourdit dans une certaine mesure la procédure. Néanmoins, si l'utilisateur ne dispose pas d'une estimation réaliste de son point de départ et/ou s'il désire un minimum global de sa fonction et/ou s'il veut un contrôle rapide de ses résultats, nous lui conseillons d'utiliser une telle approche qui sera nettement plus fiable et moins coûteuse en ressources humaines ( gain de temps ).

## Chapitre 3

### 3. Optimisation d'une fonction réelle de plusieurs variables réelles.

#### 3.0. Introduction.

Ce chapitre est consacré à l'optimisation d'une fonction réelle de plusieurs variables réelles. La première partie est consacrée à l'optimisation sans contrainte. La méthode classique reprise pour résoudre ce problème est la méthode B.F.G.S. ( Broyden, Fletcher, Goldfarb, Shanno ). Dans la deuxième partie, nous verrons comment adapter cette méthode aux problèmes avec contraintes de bornes.

Les caractéristiques essentielles de chacun de ces algorithmes sont passées en revue. Nous y introduisons ensuite l'élément graphique et terminons par une série d'exemples.

### 3.1. Méthode quasi-Newton pour les problèmes d'optimisation sans contrainte.

#### 3.1.0. Introduction.

Cette première partie du chapitre concerne les problèmes du type.

$$\begin{aligned} &\text{Minimiser } F(x) \\ &x \in \mathbb{R}^n \end{aligned}$$

où  $F(x)$  est une fonction de  $\mathbb{R}^n$  à valeur dans  $\mathbb{R}$  deux fois continûment différentiable.

Nous désignerons le gradient de la fonction par la lettre  $g$  minuscule et le hessien par la lettre  $G$  majuscule :

$$\nabla_x f(x) = g(x) \quad \text{et} \quad G(x) = \partial^2 F / \partial x_i \partial x_j$$

L'utilisation de méthodes quasi-Newton pour résoudre de tels problèmes a reçu un haut degré de popularité. depuis le début des années soixantes.

Leur caractéristique fondamentale est qu'elles n'exigent pas le calcul explicite de la matrice hessienne mais la construction d'une matrice d'approximation du hessien utilisant l'information récoltée à chaque itération sur la courbure de la fonction. De plus, lorsque nous travaillons avec une approximation  $H$  de l'inverse  $G^{-1}(x)$  du Hessien, les méthodes Quasi-Newton évitent à chaque itération la résolution d'un système d'équations linéaires pour trouver la direction

de recherche  $p^{(k)}$ , et lorsque nous utilisons une approximation  $B^{(k)}$  du Hessien le système d'équations linéaires peut être aisément résolu si nous possédons une factorisation du hessien.

La direction de recherche  $p^{(k)}$  à la  $k^{\text{ième}}$  itération est choisie par analogie avec la méthode de Newton.

$$\begin{aligned} p^{(k)} &= - H^{(k)} g(x^{(k)}) \\ &= - H^{(k)} g^{(k)} \end{aligned}$$

où  $H^{(k)}$  est en quelque sorte une approximation de l'inverse  $G^{-1}(x)$  de la matrice hessienne. Par contre, si nous travaillons avec une approximation  $B^{(k)}$  du hessien, le système à résoudre pour obtenir  $p^{(k)}$  est le suivant :

$$B^{(k)} p^{(k)} = - g^{(k)}$$

Si la matrice  $H^{(k)}$  ( $B^{(k)}$ ) est définie positive, alors  $p^{(k)}$  est une direction de descente puisque

$$p^{(k)T} g^{(k)} = - g^{(k)T} H^{(k)} g^{(k)} < 0$$

$$(-p^{(k)T} B^{(k)} p^{(k)}) = -p^{(k)T} g^{(k)} < 0$$

D'où l'exigence naturelle dans les méthodes Quasi-Newton de posséder à chaque itération une approximation définie positive du hessien ou de son inverse.

La  $(k+1)^{\text{ième}}$  approximation du hessien ou de son inverse est obtenue par une correction de rang deux, qui théoriquement garantit que la nouvelle approximation reste définie positive. Mais il arrive en pratique que même si les formules de modification garantissent le caractère défini positif de la matrice, la nouvelle approximation du hessien devienne non-définie. Ceci résulte uniquement des erreurs d'arrondis.

En résumé, la  $k^{\text{ième}}$  étape d'une méthode quasi-Newton se présente de la manière suivante :

# 1: soient  $x^{(k)}$  et  $g^{(k)}$ , respectivement l'approximation du minimum et le gradient de  $F(x)$  évalué en  $x^{(k)}$   
Calculer  $p^{(k)}$ , direction de descente, solution de l'équation :

$$B^{(k)} p^{(k)} = -g^{(k)}$$

# 2: soit à déterminer  $\alpha^{(k)}$  de manière à ce que

$$F(x^{(k)} + \alpha^{(k)} p^{(k)}) = \min_{\alpha} F(x^{(k)} + \alpha p^{(k)})$$

C'est-à-dire effectuer une recherche linéaire le long de  $p^{(k)}$  pour obtenir le point



$$x^{(k+1)} = x^{(k)} + \alpha^{(k)} p^{(k)}$$

$$\text{On posera } g^{(k+1)} = g(x^{(k+1)}).$$

#3: Modification du Hessien par une correction de rang deux.

$$B^{(k+1)} = B^{(k)} + \lambda z^{(k)} z^{(k)T} + \mu w^{(k)} w^{(k)T}$$

où  $z^{(k)}$  et  $w^{(k)}$  sont des  $n$ -vecteurs et  $\lambda$  et  $\mu$  sont des scalaires.

Les valeurs exactes de ces variables qui sont fonction de  $g^{(k)}$ ,  $p^{(k)}$ ,  $g^{(k+1)}$  dépendent de la règle de modification de l'approximation du hessien utilisée.

### 3.1.1. Description de l'algorithme de Broyden, Fletcher, Gólfarb, Shanno.

Soit le problème (P),

$$\text{minimiser } F(x) \text{ où } F \in C^2 \text{ et } x \in \mathbb{R}^n$$

Soient  $g(x)$  et  $G(x)$  respectivement le gradient et le hessien de  $F(x)$ .

Considérons le développement de Taylor limité au second ordre de  $F(x)$  autour du point  $x^{(k)}$ .

$$F(x^{(k)} + h) = F(x^{(k)}) + g(x^{(k)})^T h + \frac{1}{2} h^T G(x^{(k)}) h \quad (1)$$

Le sens de la courbure de la fonction  $F$  au point  $x^{(k)}$  est donné par l'expression  $h^T G(x^{(k)}) h$ . Si nous différencions (1) au point  $x^{(k)}$ , nous obtenons :

$$g(x^{(k)} + h) = g(x^{(k)}) + G(x^{(k)}) h$$

D'où la "courbure" de la fonction au point  $x^{(k)}$  dans la direction  $h$  peut être approximée au premier ordre par l'expression

$$h^T G(x^{(k)}) h = (g(x^{(k)} + h) - g(x^{(k)}))^T h \quad (2)$$

vérifiée  $\forall h \in \mathbb{R}^n$

Supposons qu'à la  $k^{\text{ième}}$  itération de la méthode, nous possédions un point  $x^{(k)}$  et une approximation du hessien  $B^{(k)}$  qui soit le reflet des informations récoltées sur la courbure de la fonction jusqu'à ce stade de l'optimisation et que la matrice  $B^{(k)}$  soit symétrique et définie positive.

Si nous considérons la matrice  $B^{(k)}$  comme le hessien d'une fonction quadratique,

$$Q(x) = F(x^{(k)}) + g(x^{(k)})^T x + \frac{1}{2} x^T B^{(k)} x,$$

la direction de recherche  $p^{(k)}$  est solution du système linéaire

$$B^{(k)} p^{(k)} = -g^{(k)} \quad (3)$$

qui est l'analogue de celui de la méthode de Newton décrite au chapitre 2.

En effectuant une recherche linéaire le long de la direction  $p^{(k)}$ , nous obtenons

$$x^{(k+1)} = x^{(k)} + s^{(k)} \quad \text{où} \quad s^{(k)} = \alpha^{(k)} p^{(k)}$$

et

$$g^{(k+1)} = g(x^{(k+1)})$$

Cette estimation du gradient va nous permettre de définir une nouvelle approximation du hessien  $G(x)$ . Cette nouvelle approximation du hessien, soit  $B^{(k+1)}$ , devra posséder les propriétés suivantes :

- être symétrique et définie positive.
- prendre en considération l'information sur la courbure de la fonction le long de  $s_k$ . C'est-à-dire  $B^{(k+1)}$  devra vérifier l'équation (2).

$$B^{(k+1)} s^{(k)} = g(x^{(k)} + s^{(k)}) - g(x^{(k)}) = y^{(k)} \quad (4)$$

L'équation (4) est la formulation quasi-Newton du problème.

- Etre "proche" de  $B^{(k)}$ , l'approximation précédente du hessien.

Notre problème revient alors à trouver  $B^{(k+1)}$  symétrique et définie positive minimisant l'expression

$$\|B - B^{(k)}\|_F$$

Sous la contrainte d'égalité (5)

$$B s_k = y_k$$

(  $\|\cdot\|_F$  désigne la norme de Frobénius :  $\|A\|_F = \left( \sum_{i=1}^n \sum_{j=1}^n a_{ij}^2 \right)^{\frac{1}{2}}$  )

Malheureusement, le problème n'a généralement pas de solution car l'ensemble des matrices  $\left\{ B \mid B \text{ est définie positive} \right\}$  n'est pas fermé.

Par contre, le problème plus général, qui laisse tomber la condition symétrique et définie positive, soit

$$\min_c \|c - J\|_F$$

sous la contrainte d'égalité

$$cs = y$$

où  $J, s, y$  sont donnés et  $s \neq 0$

admet une solution unique donnée par

$$c^* = J + \frac{(y - Js)s^T}{s^T s} \quad (6)$$

Démontrons cette proposition:

Soit  $c$  tel que  $cs = y$

Il vient :

$$\begin{aligned} \|c^* - J\|_F &= \left\| \frac{(y - Js)s^T}{s^T s} \right\|_F = \left\| \frac{c - J}{s^T s} ss^T \right\|_F \\ &\leq \|c - J\|_F \left\| \frac{ss^T}{s^T s} \right\|_2 \quad (\text{car } \|AB\|_F \leq \|A\|_F \|B\|_2) \end{aligned}$$

et comme

$$\left\| \begin{pmatrix} s & s^T \\ s^T & s \end{pmatrix} \right\|_2 \leq \frac{\|s\| \|s^T\|}{s^T s} = 1$$

on déduit l'inégalité

$$\|c^* - J\|_F \leq \|c - J\|_F$$

et le fait que  $c^*$  fournit une solution au problème

L'unicité est évidente,  $\|\cdot\|_F$  étant une fonction strictement convexe.

Etant donné que  $B^{(k)}$  est symétrique, définie positive si et seulement si il existe  $J^{(k)}$  non singulière telle que  $B^{(k)} = J^{(k)} J^{(k)T}$ , notre problème revient à trouver

$$J^{(k+1)} = \min_J \left\| J - J^{(k)} \right\|_F$$

Sous la contrainte d'égalité (7)

$$J v^{(k)} = y^{(k)} \quad \text{où} \quad v_k = J^{(k+1)T} s^{(k)}$$

De (6) nous déduisons  $J^{(k+1)}$  en fonction de  $v^{(k)}$

$$J^{(k+1)} = J^{(k)} + \frac{(y^{(k)} - J^{(k)} v^{(k)}) v^{(k)T}}{v^{(k)T} v^{(k)}} \quad (8)$$

En combinant (7) et (8), il vient

$$v(k) = J(k)^T s(k) + v(k) \frac{(y(k) - J(k)v(k))^T}{v(k)^T v(k)} s(k)$$

Cette égalité ne peut être vérifiée que si

$$v(k) = \alpha(k) J(k)^T s(k) \text{ où } \alpha(k) \in \mathbb{R}$$

$$\begin{aligned} \Rightarrow \alpha(k) J(k)^T s(k) &= J(k)^T s(k) + \alpha(k) J(k)^T s(k) \cdot \\ &\frac{(y(k) - J(k) \alpha(k) J(k)^T s(k))^T}{\alpha(k)^2 s(k)^T J(k) J(k)^T s(k)} s(k) \end{aligned} \quad (9)$$

de la relation (9) on déduit

$$\alpha(k)^2 = \frac{y(k)^T s(k)}{s(k)^T J(k) J(k)^T s(k)}$$

et donc

$$v(k) = \left[ \frac{y(k)^T s(k)}{s(k)^T J(k) J(k)^T s(k)} \right]^{\frac{1}{2}} J(k)^T s(k)$$

en multipliant  $J^{(k+1)}$  par  $J^{(k+1)T}$  on obtient la formule B.F.G.S. recherchée

$$B^{(k+1)} = B(k) + \frac{y(k) y(k)^T}{y(k)^T s(k)} - \frac{B(k) s(k) s(k)^T B(k)}{s(k)^T B(k) s(k)} \quad (10)$$



Si la direction de recherche est obtenue par l'équation (3) alors nous avons que

$$B^{(k)} s^{(k)} = -\alpha^{(k)} g^{(k)}$$

et l'équation (10) devient

$$B^{(k+1)} = B^{(k)} + \frac{y^{(k)} y^{(k)T}}{\alpha^{(k)} y^{(k)T} p^{(k)}} - \frac{g^{(k)} g^{(k)T}}{g^{(k)T} p^{(k)}} \quad (11)$$

La section suivante est consacrée à la résolution du système d'équations linéaires pour trouver la direction de recherche  $p^{(k)}$  en utilisant la factorisation de Cholesky de  $B^{(k)}$ .

Par conséquent, à chaque itération, la matrice d'approximation du Hessien  $B^{(k)}$  doit être factorisée. L'exploitation de la structure particulière de l'équation (11), c'est-à-dire du fait que  $B^{(k+1)}$  est une correction de rang deux de  $B^{(k)}$ , permet d'obtenir les facteurs de Cholesky de  $B^{(k+1)}$  par une modification de ceux de  $B^{(k)}$  au lieu d'effectuer l'évaluation de  $B^{(k+1)}$  puis de calculer sa factorisation de Cholesky. Nous développerons cette modification dans la section 3.1.1.2.

### 3.1.1.1. Calcul de la direction de descente.

Dans la méthode B.F.G.S., comme dans toute méthode quasi-Newton, nous devons résoudre le système d'équations linéaires.

$$B^{(k)} P^{(k)} = -g^{(k)}$$

Pour trouver la direction de recherche  $P^{(k)}$ ,  $B^{(k)}$  étant définie positive, la résolution de ce système sera facilement effectuée au moyen de la factorisation de Cholesky de  $B^{(k)}$  (obtenue aisément par les algorithmes décrits dans la section 3.1.1.2.).

$$\text{Soit } B^{(k)} = L^{(k)} D^{(k)} L^{(k)T}$$

Où  $L^{(k)}$  est une matrice triangulaire inférieure de diagonale unité

et  $D^{(k)}$  est une matrice diagonale

Il suffit alors de résoudre successivement les systèmes

$$L^{(k)} V = -g^{(k)}$$

$$L^{(k)} P^{(k)} = D^{(k)^{-1}} V$$

ce qui s'écrit encore

$$V_i = -g_i^{(k)} - \sum_{j=1}^{i-1} l_{ij}^{(k)} V_j$$

et 
$$P_i^{(k)} = V_i / d_i^{(k)} - \sum_{j=i+1}^n l_{ji}^{(k)} P_j^{(k)}$$

### 3.1.1.2. Modification des facteurs de Cholesky.

Cette section concerne la modification des facteurs de Cholesky d'une matrice définie positive  $A$  après une correction de rang un.

En terme mathématique, le problème est de calculer les facteurs de Cholesky  $\bar{L}$   $\bar{D}$   $\bar{L}^T$  tel que

$$\bar{A} = \bar{L} \bar{D} \bar{L}^T = A + \alpha z z^T = L D L^T + \alpha z z^T$$

Nous supposons en plus que les éléments  $d_j$  et  $\bar{d}_j$  sont positifs ce qui implique que  $A$  et  $\bar{A}$  sont définie positive dans le cas contraire les algorithmes itératifs tels que décrits en (11) sont numériquement instables.

Dans les méthodes quasi-Newton, il suffit d'appliquer deux fois ou de combiner les algorithmes présentés dans cette section pour obtenir les facteurs de Cholesky de la correction de rang deux de la  $k^{\text{ième}}$  approximation du Hessien, c'est à dire pour obtenir les facteurs de Cholesky de la  $(k+1)^{\text{ième}}$  approximation du hessien.

Nous avons que

$$\begin{aligned} \bar{A} &= L D L^T + \alpha z z^T \\ &= L ( D + \alpha P P^T ) L^T \end{aligned}$$

où  $P$  est solution des équations

$$L P = z$$

si nous calculons les facteurs de Cholesky de  $D + \alpha PP^T$

$$D + \alpha PP^T = \tilde{L} \tilde{D} \tilde{L}^T$$

nous obtenons les facteurs de Cholesky de  $A$  qui sont de la forme

$$\bar{A} = L \tilde{L} \tilde{D} \tilde{L}^T L^T$$

En posant  $\bar{L} = L \tilde{L}$  et  $\bar{D} = \tilde{D}$ , nous obtenons  $\bar{A} = \bar{L} \bar{D} \bar{L}^T$  et  $\bar{L}$  est une matrice triangulaire inférieure de diagonale unité, car le produit de deux matrices triangulaires inférieures de diagonale unité est une matrice triangulaire inférieure de diagonale unité.

Première méthode. Gill, Golub, Murray et Saunders ( 1974 )

. La factorisation de Cholesky de  $D + \alpha PP^T$  peut être obtenue directement en utilisant les équations de récurrence habituelles démontrées au lemme 1 de l'annexe 1.

$$D + \alpha PP^T = \tilde{L} \tilde{D} \tilde{L}^T$$

$$\text{où} \quad \tilde{d}_j + \sum_{i=1}^{j-1} \tilde{d}_i \tilde{l}_{ji}^2 = d_j + P_j^2 \quad (12)$$

$$\text{et} \quad \sum_{i=1}^{j-1} \tilde{d}_i \tilde{l}_{ji} \tilde{l}_{ri} + \tilde{d}_j \tilde{l}_{rj} = \alpha P_i P_r, r=j+1 \dots$$

De plus

. L est une matrice triangulaire inférieure spéciale de paramètres B,P,1 c'est-à-dire une matrice du type (cfr lemme 2, de l'annexe 1 )

$$\begin{pmatrix} 1 & & & & \\ P_2 B_1 & 1 & & & \\ P_3 B_1 & P_3 B_2 & 1 & & \\ \vdots & \vdots & & \ddots & \\ P_n B_1 & P_n B_2 & \dots & P_n B_{n-1} & 1 \end{pmatrix}$$

$$\text{où } B_i = \frac{P_i}{\tilde{d}_i} \left( \alpha - \sum_{i=1}^{j-1} \tilde{d}_i B_i^2 \right) \quad (14)$$

et les éléments de  $\tilde{D}$  sont donnés par

$$\tilde{d}_j = d_j + P_j^2 \left( \alpha - \sum_{i=1}^{j-1} \tilde{d}_i B_i^2 \right) \quad (15)$$

Comme  $\tilde{L}$  est une matrice triangulaire inférieure spéciale le produit  $L \tilde{L}$  s'obtient par les relations de récurrence suivantes déduite du lemme 3 de l'annexe 1

$$1) \text{ Résoudre } LP = z^{(1)} \quad (16)$$

2) Pour  $j = 1, 2, \dots, n$

$$\bar{L}_{jj} = l_{jj} = 1$$

$$\left. \begin{aligned} z_r^{(j+1)} &= z_r^{(j)} - P_j l_{rj} \\ \bar{L}_{rj} &= l_{rj} + B_j z_r^{(j+1)} \end{aligned} \right\} \quad r=j+1, \dots, n \quad (17)$$

En utilisant la relation (14), (15), (16), (17), en

posant 
$$\alpha_j = \alpha - \sum_{i=1}^{j-1} \bar{d}_i B_i^2$$

et en remarquant que

$$z_j^{(j)} = z_j - \sum_{i=1}^{j-1} l_{ji} P_i = P_j$$

on dérive l'algorithme de calcul :

1) Définir  $\alpha_1 = \alpha$  et  $W^{(1)} = z$

2) Pour  $j = 1, 2, \dots, n$

$$P_j = W_j^{(j)}$$

$$\bar{d}_j = d_j + \alpha_j P_j^2$$

$$B_j = P_j \alpha_j / \bar{d}_j$$

$$\alpha_{j+1} = d_j \alpha_j / \bar{d}_j \quad (18)$$



$$\left. \begin{aligned} W_r^{(j+1)} &= W_r^{(j)} - P_j l_{rj} \\ \bar{l}_{rj} &= l_{rj} + B_j W_r^{(j+1)} \end{aligned} \right\} r = j+1, \dots, n \quad (19)$$

La relation (18) s'obtient simplement :

$$\begin{aligned} \alpha_{j+1} &= \alpha_j - d_j B_j^2 = \alpha_j - \bar{d}_j P_j^2 \alpha_j^2 / \bar{d}_j^2 = \alpha_j (1 - \bar{d}_j P_j^2 / \bar{d}_j^2) \\ &= \alpha_j (1 - \alpha_j P_j^2 / \bar{d}_j) \end{aligned}$$

Comme  $\bar{d}_j = d_j + \alpha_j P_j^2$ , nous obtenons  $\alpha_{j+1} = \alpha_j d_j / \bar{d}_j$   
 Quant aux équations (19) on peut les transformer de la manière  
 Suivante :

$$\bar{l}_{rj} = l_{rj} + B_j (W_r^{(j)} - P_j l_{rj})$$

$$\bar{l}_{rj} = (1 - B_j P_j) l_{rj} + B_j W_r^{(j)}$$

or  $B_j = P_j \alpha_j / \bar{d}_j$  d'où  $\alpha_j = B_j \bar{d}_j / P_j$

et comme  $\bar{d}_j = d_j + \alpha_j P_j^2$

nous obtenons :

$$\bar{d}_j = d_j + B_j \bar{d}_j P_j$$

(=)  $(1 - B_j P_j) = d_j / \bar{d}_j$

d' où les relations (19) peuvent être réécrites

$$\left. \begin{aligned} \bar{l}_{rj} &= (d_j / \bar{d}_j) l_{rj} + B_j W_r^{(j)} \\ w_r^{(j+1)} &= W_r^{(j)} - P_j l_{rj} \end{aligned} \right\} r = j+1..n \quad (20)$$

Nous obtenons ainsi la forme de l'algorithme de Gill, Golub, Murray et Saunders

En comparant les expressions (19) et (20), on constate un accroissement du nombre d'opérations ( multiplications ) de 50 %. Bien que plus coûteux en temps, l'algorithme (20) est cependant plus stable. En effet, l'analyse de Fletcher et Powell (1973) a montré que l'erreur d'arrondi dépendait dans le cas (19) de  $\bar{d}_j / d_j$  et dans le cas (20) de l'inverse de cette expression, c'est à dire de  $d_j / \bar{d}_j$ .

Ainsi si  $\alpha_j$  est positif, les relations  $\bar{d}_j = d_j + \alpha_j P_j^2$   
 $\Rightarrow \bar{d}_j > d_j \quad \forall j$  et il s'ensuit une plus grande stabilité du second algorithme. Gentleman (1973) a suggéré d'utiliser les formules (19) jusqu'à un rapport fixe de  $\bar{d}_j / d_j$ . Ainsi nous combinons la rapidité et la précision dans l'algorithme.

Si  $\alpha < 0$  et  $\bar{A}$  est proche d'une singularité, il est possible que l'erreur d'arrondi provoque l'annulation d'un élément  $\bar{d}_j$  de la diagonale. Dans de tels cas il est aussi possible que  $\bar{d}_j$  change de signe, même si théoriquement, la modification engendre une matrice symétrique, définie positive.

Il est avantageux, dans ce cas, d'utiliser la méthode suivante proposée par Gill, Murray et Saunders (1975).

Deuxième Méthode : Gill , Murray et Saunders.

$$\text{Soit } \bar{A} = \bar{L} \bar{D} \bar{L}^T = A - \alpha z z^T = L D L^T - \alpha z z^T \quad \alpha > 0$$

Nous pouvons réécrire le problème sous la forme

$$\bar{A} = \bar{L} \bar{D} \bar{L}^T = L D L^T - v v^T$$

Ce qui nécessite  $n$  divisions et une racine carrée et minimise la probabilité d'"overflow et "underflow" quand  $\alpha$  est grand et la  $\|z\|$  est petite.

$$\bar{A} = L ( D - P P^T ) L^T \quad (*)$$

$$\text{avec } LP = v$$

Considérons la quantité  $\alpha^2 = 1 - P^T D^{-1} P$

De (\*) nous avons que

$$\det (\bar{A}) = (\det (L))^2 \det (D - PP^T)$$

Comme  $L$  est une matrice triangulaire inférieure de diagonale unité nous avons que le  $\det (L) = 1$   
d'où  $\det (\bar{A}) = \det (D - PP^T) = \alpha^2 \det (D)$

En supposant que  $\bar{A}$  est définie positive,  $\det(\bar{A}) > 0$  et donc  $\alpha^2 > 0$ ;

Ceci nous permet d'appliquer la proposition 1 de l'annexe 1 pour obtenir la factorisation de  $D-PP^T = M D M^T$  en utilisant les relations de récurrence suivantes

$$(1) \quad t_{m+1} = \alpha^2$$

$$(2) \quad \text{Pour } j = m, m-1, \dots, 1$$

$$t_j = t_{j+1} + P_j^2 \alpha_j \quad (21)$$

$$\bar{d}_j = d_j \quad t_{j+1} / t_j$$

$$B_j = - P_j / (d_j \quad t_j + 1)$$

Comme les éléments du vecteur B sont calculés dans l'ordre  $B_m, B_{m-1}, \dots, B_1$ , il est donc préférable d'utiliser la récurrence du lemme 4 de l'annexe 1 pour calculer le produit LM

C'est à dire

$$\text{Pour } j = m, m-1 \dots 1$$

$$V_j^{(j)} = P_j$$

$$\left. \begin{aligned} \bar{l}_{rj} &= l_{rj} + B_j V_r^{(j+1)} \\ v^{(i)} &= v_r^{(j+1)} + P_j l_{rj} \end{aligned} \right\} r = j+1 \dots n \quad (22)$$

Comme mentionné plus haut l'erreur en utilisant les relations (22) est multipliée par le facteur  $\bar{d}_i/d_j$  et  $\bar{d}_i \leq d_j$  pour tout  $j$ . Malheureusement contrairement à l'algorithme précédent, la formation de  $\bar{L}$  exige la connaissance du vecteur  $P$ , avant de commencer les relations de récurrence, pour trouver le vecteur  $B$ . Le trait fondamental de cette méthode est que si  $\alpha^2 > 0$ , la matrice modifiée est toujours définie positive malgré les erreurs d'arrondi, grâce aux relations (21).

L'algorithme final calculant (\*) est donc le suivant

(1) résoudre  $LP = V$

$$\text{définir } t_{m+1} = \alpha^2 = 1 - P^T D^{-1} P$$

Si  $t_{m+1} \leq 0$  alors  $t_{m+1} = \varepsilon$  ( $\varepsilon > 0$ ; précision machine).

(2) Pour  $j = m, m-1, \dots, 1$

$$t_j = t_{j+1} + P_j^2 / d_j$$

$$\bar{d}_j = d_j t_{j+1} / t_j$$

$$B_j = -P_j / (d_j t_{j+1})$$

$$v_j^{(j)} = P_j$$

$$\left. \begin{aligned} \bar{l}_{rj} &= l_{rj} + B_j v_r^{(j+1)} \\ v_r^{(j)} &= v_r^{(j+1)} + P_j l_{rj} \end{aligned} \right\} r = j+1 \dots m$$

### 3.1.2. Implémentation graphique.

Le logiciel d'aide graphique à l'optimisation OPTIGRAPH que nous avons développé ( voir chapitre 4 ) requiert dans le cas de minimisation sans contrainte les informations suivantes :

- x : le point courant.
- g : le gradient de la fonction au point courant
- P : la direction de descente.
- H : le hessien ou une approximation du hessien au point courant. Il peut être communiqué à OPTIGRAPH sous deux formes, soit factorisée, ( Cholesky ) soit non factorisée.

OPTIGRAPH permet alors de visualiser la fonction et l'approximation quadratique passant par le point courant. Le logiciel est particulièrement bien adapté pour les méthodes du type "Newton" ou "quasi-Newton" en dimension supérieure ou égale à deux.

Dans le cas où la dimension du problème est strictement supérieure à deux, l'utilisateur peut visualiser la fonction ou l'approximation quadratique de celle-ci dans un plan de son choix. Il dispose ainsi d'une aide efficace pour:

- choisir le point initial
- suivre le point courant ( itéré successif )
- contrôler le résultat.



Dans la section 3.3., nous reviendrons sur ces problèmes.

Si la dimension du problème est égale à deux, OPTIGRAPH définit le plan de visualisation par les équations paramétriques suivantes :

$$y_1 = x_1 + \alpha$$

$$y_2 = x_2 + \mu$$

C'est-à-dire<sup>qu'</sup> il prend comme origine de son système d'axe le point courant.

Par contre si la dimension du problème est supérieure à deux, l'utilisateur a le libre choix des plans de visualisation.

Les plans que lui propose OPTIGRAPH, en standard, sont les suivants:

- le plan des directions de descente.
- le plan des deux plus grandes valeurs propres.
- le plan de la plus grande et de la plus petite valeur propre.

Dans le cas d'une méthode quasi-Newton, le plan des directions de descente correspond au plan passant par le point courant et défini par le gradient et la direction Quasi-Newton.



En d'autres termes, il est généré par la direction de la plus grande pente ou sens courant et par la direction de la plus forte pente ou sens de la norme.

$$\|y\|^2 = y^T H Y.$$

Ce plan est particulièrement riche en informations sur les conditions de la minimisation. Une analyse graphique du comportement de la fonction dans ce plan permet à l'utilisateur "d'interagir" avec l'algorithme si nécessaire.

Le plan des valeurs propres nous donne, quant à lui, une idée du conditionnement du problème dans le voisinage du point courant et de celui de l'approximation du hessien. (visualisation de l'approximation quadratique). Il permet en plus de trouver une direction privilégiée pour relancer la minimisation si nécessaire.

### 3.1.3. Exemples.

#### 3.1.3.1. Premier cas d'application.

Soit à minimiser la fonction

$$F(X) = 100 \left( (X_3 - 10 \theta)^2 + (r - 1)^2 \right) + X_3^2$$

$$\text{où } r = (X_1^2 + X_2^2)^{\frac{1}{2}}$$

$$\text{et } 2\pi\theta = \begin{cases} \arctan(X_2/X_1) & X_1 > 0 \\ \pi + \arctan(X_2/X_1) & X_1 < 0 \end{cases}$$

Soit  $X^0 = (5, 5, 5)$  les figures 1, 2 et 3 représentent respectivement une coupe de la fonction dans les plans  $(X_1, X_3)$ ,  $(X_2, X_3)$ ,  $(X_1, X_2)$ .

Grâce à ces trois représentations nous pouvons localiser le minimum à l'aide des équations paramétriques des différents plans.

Le point à partir duquel on pourrait lancer l'optimisation est obtenu de la manière suivante :

Les équations paramétriques du plan  $(X_2, X_3)$  sont

$$X_1 = 5$$

$$X_2 = 5 + \alpha$$

$$X_3 = 5 + \beta$$

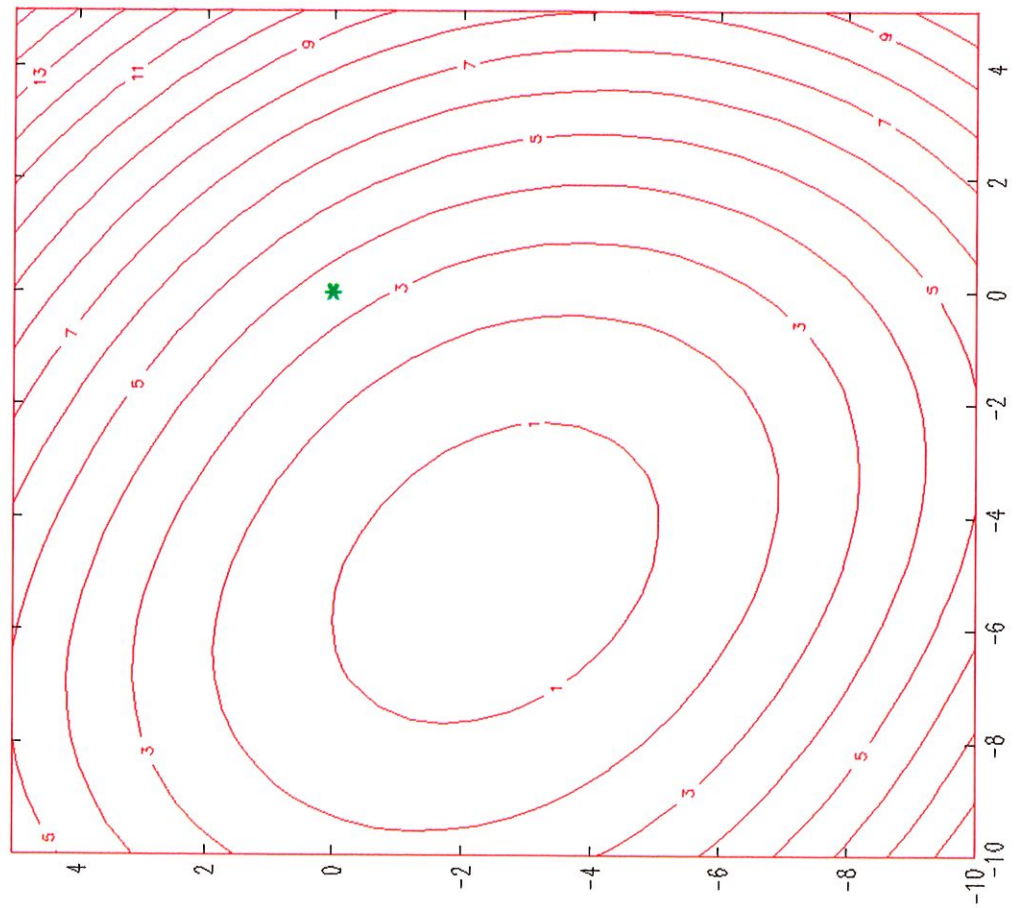
A l'aide du graphisme (Figure 2) nous constatons que les valeurs de paramètre et approximant le minimum sont  $-5$  pour  $\alpha$  et  $-5$  pour  $\beta$ . En les remettant dans les équations paramétriques on obtient  $X_2 = 0$  et  $X_3 = 0$ . En procédant de la sorte dans le plan  $(X_1, X_2)$  par exemple, on obtiendra pour  $X_1$  la valeur 2.

Dans le cas où le point initial est  $(5,5,5)$ , la figure 4 donne une vue isométrique générale de la fonction dans le plan  $(X_2, X_3)$  et les figures 5 et 6 fournissent une visualisation de la fonction dans le plan des directions de descente lors de la première itération. La figure 5 nous montre deux minima et la figure 6 nous assure que le point courant va "converger" vers le plan.

Le minimum global de la fonction est en effet le point  $(1,0,0)$

# ITERATION O

PLAN DE LA + GRANDE ET LA + PETITE VAL. PROPRE



CONTOUR KEY	
1	2192.6
2	3354.7
3	4516.7
4	5678.8
5	6840.9
6	8002.9
7	9165.0
8	10327.1
9	11489.1
10	12651.2
11	13813.2
12	14975.3
13	16137.4
14	17299.4
15	18461.5

FIGURE 1

# ITERATION 0

PLAN DES DEUX PLUS GRANDES VALEURS PROPRES

CONTOUR KEY	
1	2125.3
2	3168.4
3	4211.5
4	5254.6
5	6297.7
6	7340.8
7	8383.9
8	9427.0
9	10470.1
10	11513.2
11	12556.3
12	13599.4
13	14642.5
14	15685.6
15	16728.7

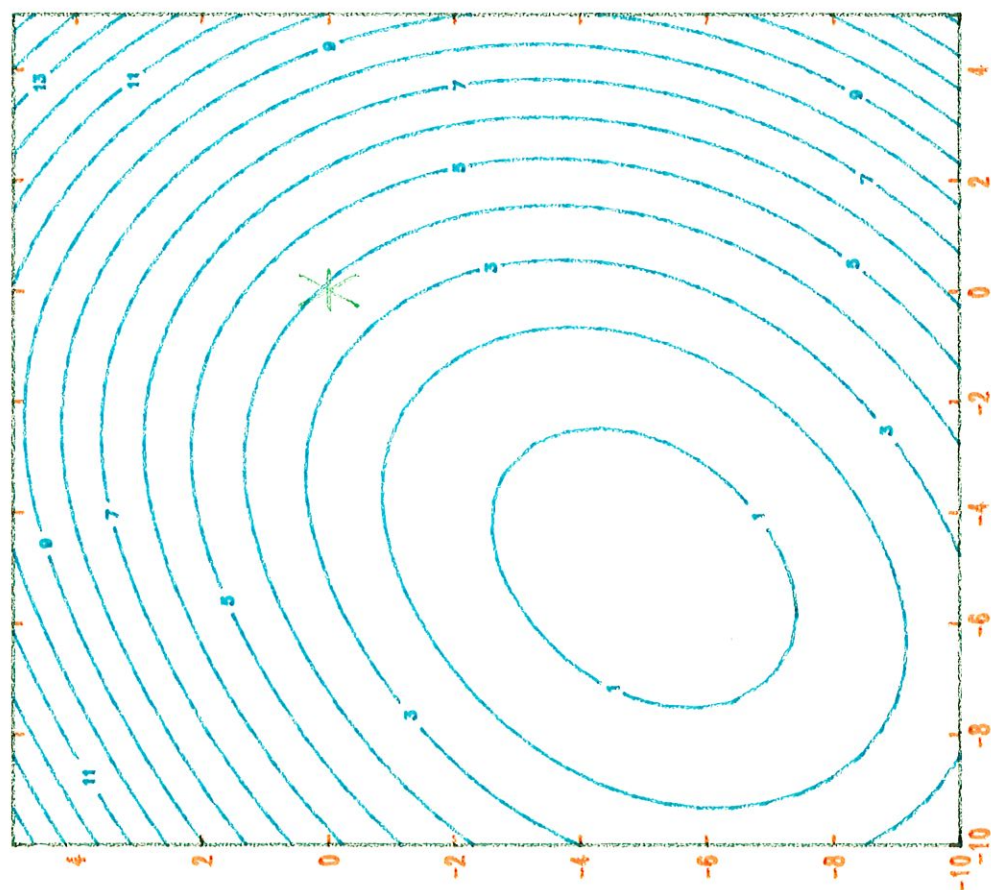
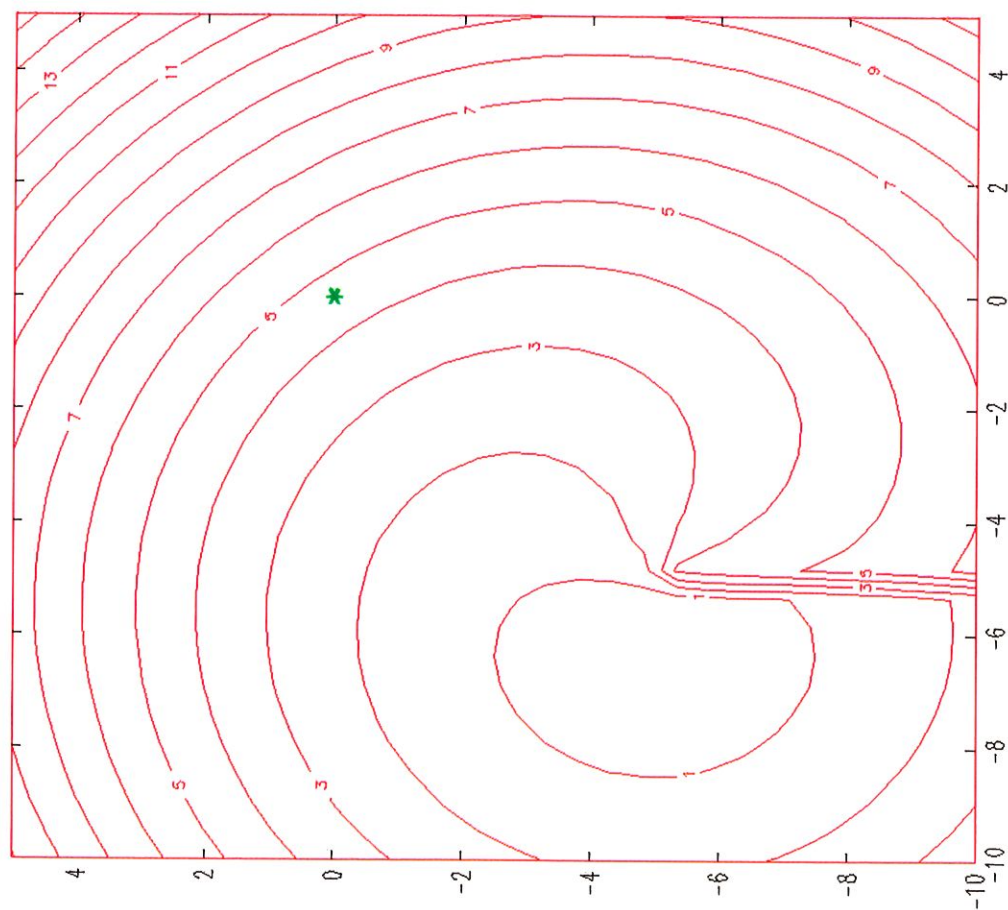


FIGURE 2



# ITERATION

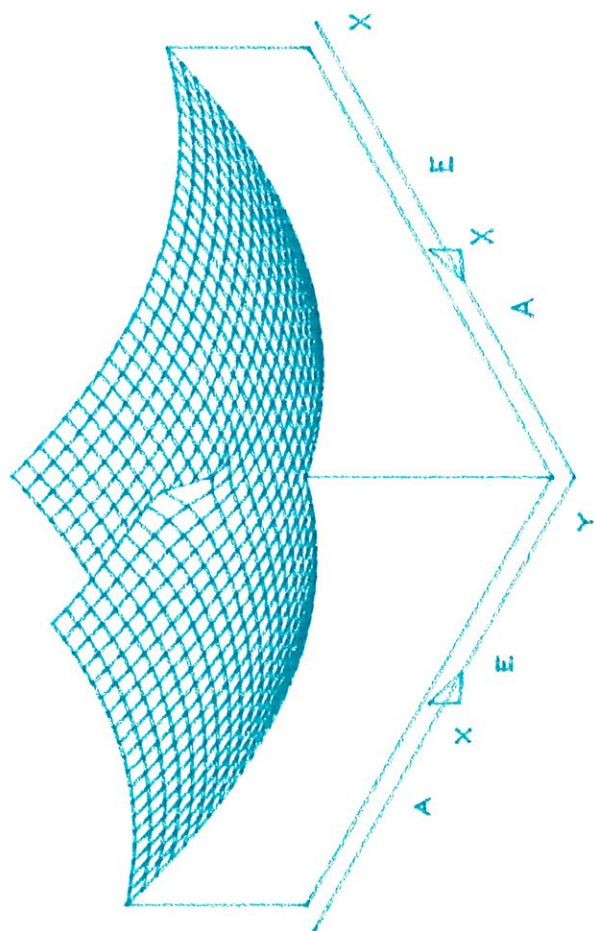
O



CONTOUR KEY	
1	658.5
2	1903.0
3	3147.4
4	4391.8
5	5636.3
6	6880.7
7	8125.1
8	9369.6
9	10614.0
10	11858.4
11	13102.9
12	14347.3
13	15591.7
14	16836.2
15	18080.6

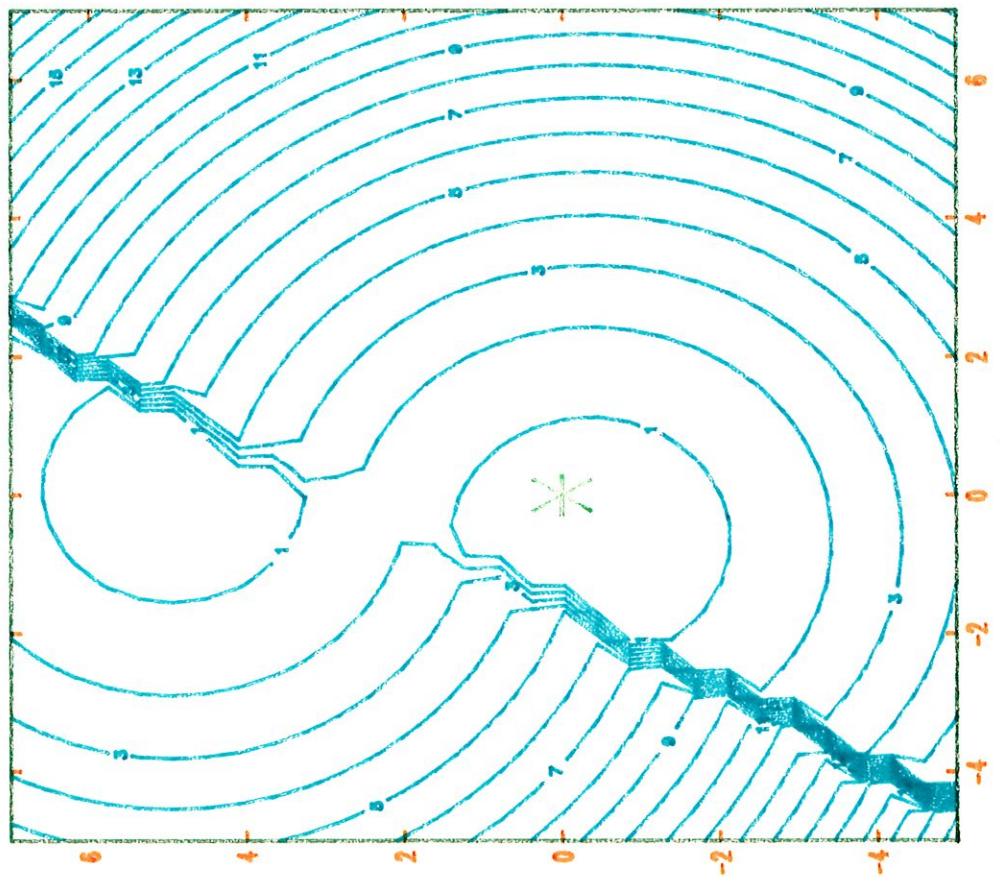
FIGURE 3

FIGURE 4





# ITERATION 1 PLAN DES DIRECTIONS DE DESCENTE

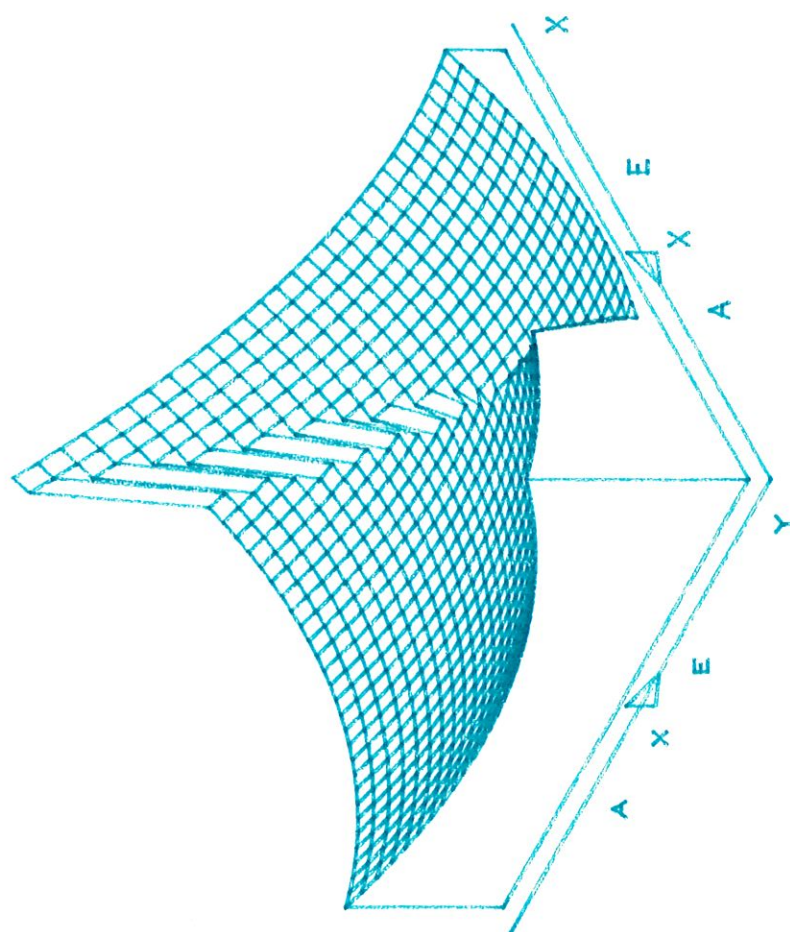


CONTOUR KEY	
1	2.2515078001231507800
2	2.2515078001231507800
3	2.2515078001231507800
4	2.2515078001231507800
5	2.2515078001231507800
6	2.2515078001231507800
7	2.2515078001231507800
8	2.2515078001231507800
9	2.2515078001231507800
10	2.2515078001231507800
11	2.2515078001231507800
12	2.2515078001231507800
13	2.2515078001231507800
14	2.2515078001231507800
15	2.2515078001231507800
16	2.2515078001231507800

FIGURE 5

# PLAN DES DIRECTIONS DE DESCENTE

FIGURE 6



### 3.1.3.2. Deuxième cas d'application.

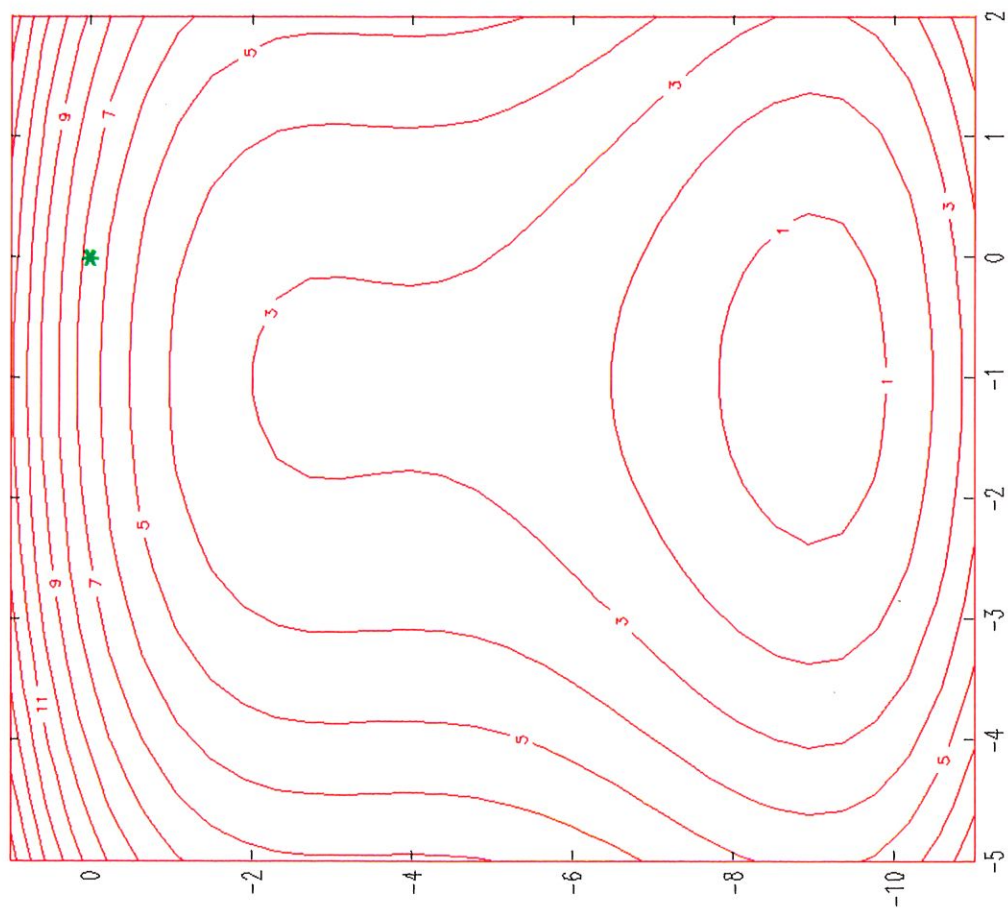
Soit à minimiser la fonction

$$F(X_1, X_2) = 100 X_1^2 - 400 X_1 + 3X_2^4 - 8X_2^3 - 54 X_2^2 + 216 X_2 + 8$$

Soit  $X^0 = (3, 6)$ , la figure 7 nous montre que le minimum global se trouve dans un voisinage du point  $(2, -3)$ . Si on laisse l'initiative à l'algorithme partant de ce point initial, il converge vers l'extrémum local  $(2, 3)$ . D'où l'intérêt de relancer la minimisation à partir d'un meilleur choix initial.

# ITERATION

O



CONTOUR KEY	
1	-878.5
2	-505.3
3	-132.1
4	241.1
5	614.3
6	987.5
7	1360.8
8	1734.0
9	2107.2
10	2480.4
11	2853.6
12	3226.8
13	3600.0
14	3973.2
15	4346.4

FIGURE 7

### 3.2. Une méthode pour résoudre les problèmes d'optimisation avec contraintes de bornes.

#### 3.2.0. Introduction.

Considérons le problème (PB) de la minimisation d'une fonction  $F(x)$ ,  $x \in \mathbb{R}^n$ , deux fois continûment différentiable, sous les contraintes de bornes :

$$l \leq x \leq u \quad \text{où} \quad l \text{ et } u \in \mathbb{R}^n$$

De tels problèmes (PB) se rencontrent fréquemment. Une méthode régulièrement utilisée pour résoudre les problèmes soumis à des contraintes non-linéaires est de les transformer en une séquence de problèmes sans contrainte. En général, les problèmes avec contraintes non-linéaires admettent aussi des contraintes de bornes et il n'est pas prouvé que l'inclusion de ces contraintes de bornes dans la fonction de pénalité permette d'aboutir à de meilleurs résultats que ceux obtenus par la résolution d'une séquence de problème de type (PB).

Par contre, si les problèmes de minimisation d'une fonction soumise uniquement à des contraintes de bornes peuvent être résolus, en général, par les algorithmes spécifiques aux contraintes linéaires, il est avantageux d'adapter ces algorithmes au cas particulier des problèmes (PB) pour obtenir des méthodes plus efficaces et plus rapides grâce à l'exploitation de la structure simple des contraintes de bornes.



### 3.2.0.1. Stratégie des contraintes actives pour les problèmes avec contraintes linéaires.

Considérons le problème (PL) de la minimisation d'une fonction deux fois continûment différentiable soumise à des contraintes linéaires.

$$(PL) : \text{Min} \quad F(x)$$

$$x \in \mathbb{R}^n$$

sous les contraintes :

$$(a_i, x) = b_i \quad i \in E$$

$$(a_j, x) \leq b_j \quad j \in E'$$

Il est évident que les problèmes du type (PB) sont des cas particuliers de celui-ci.

La stratégie des contraintes actives est basée sur la proposition suivante :

$$\begin{aligned} & \text{Soient } x^*, \text{ un point admissible de (PL) et } I(x^*) \\ & = \left\{ j \mid (a_j, x^*) = b_j, j \in E' \right\} \text{ et soit le problème (PR) :} \\ & \quad \text{Min } F(x) \\ & \quad x \in \mathbb{R}^n \end{aligned}$$

Sous les contraintes :

$$(a_i, x) = b_i \quad i \in E \quad (1)$$

$$(a_j, x) = b_j \quad j \in I(x^*) \quad (2)$$

Alors si  $(x^*, u^*_i \ (i \in E), \lambda^*_j \ (j \in I(x^*)))$  est un point de Kuhn-Tucker du problème (PR) (c'est à dire un point pour lequel le lagrangien du problème (PR) s'annule).

$$\text{Et si } \lambda^*_j \geq 0 \quad \forall j \quad j \in I(x^*) \quad (3)$$

Alors  $(x^*, u^*, \lambda^*)$  est un point de Kuhn-Tucker de (PL).

Dans la stratégie des contraintes actives nous sommes donc obligés de dissocier le problème de Kuhn-Tucker en deux étapes :

- résolution d'un problème aux contraintes d'égalités.
- recherche des multiplicateurs de Lagrange négatifs.

Il se pourrait, en effet, que certains multiplicateurs de Lagrange associés à des contraintes du type (2) soient négatifs.

Par conséquent, la méthode recherche un ensemble de contraintes actives tel qu'un minimum du problème réduit (PR) soit aussi un minimum du problème (PL).



La structure d'un algorithme fondé sur la stratégie des contraintes actives est la suivante.

. Soit  $k = 1$

1. Etant donné un point admissible  $x^{(k)}$  c'est à dire satisfaisant aux contraintes du problème (PL)  
Evaluer  $A^{(k)} = E \cup I(x)^{(k)}$ , l'ensemble des contraintes actives.

2. Si  $x^{(k)}$  n'est pas solution du problème réduit :

$$\text{Min } F(x)$$

(PR) sous les contraintes

$$(a_i, x) = b_i \quad i \in A^{(k)}$$

Calculer en  $x^{(k)}$  une direction de descente admissible  $s^{(k)}$  pour (PR) et aller en 4.

Sinon aller en 3.

3. Evaluer les multiplicateurs de Lagrange  $\lambda_i$   $i \in A^{(k)}$   
et tester si  $x^{(k)}$  est solution de (PL) ( c'est-à-dire  
si  $\lambda_i \geq 0 \quad \forall i \in A^{(k)} \cap I(x^{(k)})$  )

. si c'est le cas alors STOP.

. sinon . retirer de  $A^{(k)}$  la contrainte d'inégalité correspondant au multiplicateur  $\lambda_i$  le plus négatif.

. calculer une direction de descente admissible  $s^{(k)}$  pour le nouveau problème (PR) et aller en 4.

4. Effectuer une recherche linéaire admissible le long de  $s^{(k)}$  et poser  $x^{(k+1)} = x^{(k)} + \alpha^{(k)} s^{(k)}$

$$5. \text{ Si } \alpha^{(k)} = \alpha_{\max}^{(k)} = \begin{cases} \min \left\{ \frac{b_j - (a_j, x^{(k)})}{(a_j, s^{(k)})} \mid j \notin A^{(k)}, (a_j, s^{(k)}) > 0 \right\} \\ \text{si } \exists j: (a_j, s^{(k)}) > 0 \\ +\infty \quad \text{sinon} \end{cases}$$

Alors ajouter  $p$  à  $A^{(k)}$  où  $p$  est l'indice donnant le minimum

6.  $k \leftarrow k+1$  et aller en 2.

### 3.2.0.2. Adaptation de la stratégie des contraintes actives aux problèmes avec contraintes de bornes uniquement.

Le problème (PB) étant un cas particulier du problème (PL) nous allons utiliser la méthode des contraintes actives en exploitant la structure simple du problème (PB) pour obtenir un algorithme spécifique et donc moins général mais plus efficace pour résoudre les problèmes de minimisation soumis uniquement à des contraintes de bornes.

Les modifications de l'algorithme général de la méthode des contraintes actives sont résumées dans les trois points suivants :

- 1) Rappelons que le problème (PB) consiste à minimiser la fonction  $F(x) \in C^2$  ( $x \in \mathbb{R}^n$ ) sous les contraintes de bornes :

$$l_i = x_i = u_i \quad i \in E \quad \text{contraintes d'égalité} \quad (4)$$

$$l_j \leq x_j \leq u_j \quad j \in E' \quad \text{contraintes d'inégalité} \quad (5)$$

Les multiplicateurs de Lagrange associés aux contraintes du type (5) sont de la forme :

$$\lambda_j = \begin{cases} g_j & \text{si } x_i = l_j \\ -g_j & \text{si } x_j = u_j \\ 0 & \text{sinon} \end{cases}$$

où  $g_j$  désigne la  $j^{\text{ième}}$  composante du gradient.

2) Le pas maximum admissible est donné par :

$$\alpha_{\max}^{(k)} = \min_i \left\{ \alpha_i \right\}$$

où  $\alpha_i = \max \left\{ \frac{l_i - x_i^{(k)}}{p_i^{(k)}}, \frac{u_i - x_i^{(k)}}{p_i^{(k)}}; p_i^{(k)} \neq 0 \right\}$

3) Comme les contraintes sont uniquement des contraintes de bornes, nous obtenons que le problème (PR) :

Minimiser  $F(x)$ ,  $x \in \mathbb{R}^n$ ,  $F(x) \in \mathbb{C}^2$   
 sous les contraintes

$$l_i = x_i = u_i \quad i \in E$$

$$l_j = x_j \quad \text{ou} \quad x_j = u_j \quad j \in I(x^{(k)})$$

où  $x^{(k)}$  est la  $k^{\text{ième}}$  estimation du minimum de (PB)

$$\text{et } I(x^{(k)}) = \left\{ j \mid l_j = x_j^{(k)} \quad \text{ou} \quad x_j^{(k)} = u_j \right\}$$

est équivalent au problème :

Minimiser  $F(y)$  pour  $y \in \mathbb{R}^{n-t}$

$t$  désignant le nombre de contraintes du problème (PR);  
 c'est-à-dire équivaut à minimiser  $f(x)$  dans l'espace des  
 variables libres ( variables non-fixées sur une borne ).

Nous obtenons ainsi un problème de minimisation sans contrainte et par conséquent, nous pouvons appliquer la méthode B.F.G.S. pour le résoudre.

Ces trois résultats permettent d'obtenir l'algorithme suivant :

Soit  $x^{(k)}$  la  $k^{\text{ième}}$  approximation de la solution de (PB)

Soient  $F^{(k)}$  et  $g^{(k)}$ , respectivement la fonction et le gradient de la fonction évalué en  $x^{(k)}$  et soient  $G^{(k)}$  la  $k^{\text{ième}}$  approximation du hessien,  $z$  une  $n \times (n-t)$  matrice contenant les colonnes non-nulles de la matrice diagonale dont ses éléments diagonaux sont égaux à un si la variable correspondante est libre, et à zéro si elle est fixée sur une borne;  $t$  est le nombre de variables fixées.

Soit  $\bar{g} = z^t g^{(k)}$  le "gradient projeté"

- 1) si  $\|\bar{g}\| < \varepsilon_1$ , calculer le vecteur des multiplicateurs de Lagrange dont le  $i^{\text{ième}}$  élément est défini par
 
$$\lambda_i = \pm \hat{g}_i \text{ où } \hat{g} \text{ est le vecteur contenant les éléments de } g \text{ qui n'appartiennent pas au gradient projeté } \bar{g}. \hat{g}_i \text{ est pris avec le signe négatif si le multiplicateur correspond à une borne supérieure et avec le signe positif sinon.}$$

Soit  $s$  l'indice tel que  $\lambda_s \leq \lambda_i \quad i=1, \dots, t$

Si  $\lambda_s < -\varepsilon$  alors la variable correspondante devient libre et l'itération est recommencée avec l'ensemble des variables libres augmenté d'une unité. Dans ce cas il est nécessaire d'ajouter une ligne et une colonne aux facteurs de Cholesky de l'approximation du Hessien, ce qui se réalise de la manière suivante :

$$L^{(k)} \text{ devient } \left[ \begin{array}{c|c} L^{(k)} & 0 \\ \hline 0 & 1 \end{array} \right], \quad D^{(k)} \text{ devient } \left[ \begin{array}{c|c} D^{(k)} & 0 \\ \hline 0 & 1 \end{array} \right]$$

sinon si  $\|\bar{g}\| < \varepsilon_2$

alors une recherche linéaire spéciale est effectuée

Si la recherche découvre un "suffisamment meilleur" point  $x^{(k+1)}$

Alors une nouvelle itération est recommencée.

Sinon si  $\lambda_s > \varepsilon$ ,  $x^{(k)}$  est considérée comme solution de (PB) et l'algorithme est terminé.

Sinon la variable correspondant à  $\lambda_s$  est relâchée et l'itération est recommencée avec un nouvel ensemble de variables libres.

- 2) Calculer la direction de descente  $P$  qui est obtenue par les équations suivantes :

$$P = -z \bar{P}$$



où  $\bar{P}$  est solution du système d'équations linéaires.

$$B^{(k)} \quad \bar{P} = L^{(k)} D^{(k)} L^{(k)T} \bar{P} = - \bar{g}$$

3) Calculer le pas maximum admissible le long de  $P$

$$\alpha_i = \max \left\{ \frac{l_i - x_i^{(k)}}{P_i}, \frac{u_i - x_i^{(k)}}{P_i}; P_i \neq 0 \right\}$$

$$\alpha_{\max} = \min_i (\alpha_i)$$

et effectuer une recherche linéaire pour déterminer la longueur du pas qui doit être au maximum égale à  $\alpha_{\max}$

4) Définir la nouvelle approximation  $x^{(k+1)} = x^{(k)} + \alpha P$

5) Calculer

$$L^{(k+1)} D^{(k+1)} L^{(k+1)T} = L^{(k)} D^{(k)} L^{(k)T} + \frac{\bar{Y} \bar{Y}^T}{\alpha \bar{P}^T \bar{Y}} - \frac{\bar{g} \bar{g}^T}{\bar{P}^T \bar{g}}$$

$$\text{où } \bar{Y} = z^T (g^{(k+1)} - g^{(k)})$$

En utilisant les algorithmes de modifications des facteurs de Cholesky décrites au début du chapitre

. si  $\alpha = \alpha_{\max}$  une variable atteint une de ses bornes.

On diminue l'ensemble des variables libres et on modifie

$L^{(k+1)} D^{(k+1)} L^{(k+1)T}$  suivant la règle décrite dans la

section 3.2.0.3.



6)  $K \leftarrow K+1$  et aller en 1.

Remarques.

Dans la programmation de l'algorithme, Gill et Murray (1973) conseillent de prendre pour valeur d'  $\epsilon$  et d'  $\epsilon_2$  respectivement  $10 S$  et  $S^{\frac{1}{2}}$  où  $S$  est la précision machine.

Le troisième paramètre  $\epsilon_1$  contrôle la précision avec laquelle la fonction est minimisée dans le domaine des variables libres. Jusqu'au moment où  $\|\bar{g}\| < \epsilon_1$ , une variable fixée sur une borne ne sera pas relâchée.

Ils proposent comme valeur d'  $\epsilon_1$ ,  $10^{-2}$  qui apparaît être un bon compromis entre une variable fixée momentanément et une variable libre qui retourne sur l'une de ses bornes régulièrement.

Dans l'algorithme nous n'avons pas envisagé le cas des contraintes du type  $l_i = x_i = u_i$ . Dans ce cas à l'étape 1, nous ne tenons pas compte des multiplicateurs de Lagrange associés à ces contraintes.

### 3.2.0.3. Modifications de l'approximation du Hessien quand une variable atteint une borne.

Quand une variable libre devient fixée, soit la  $j^{\text{ième}}$  le sous espace de travail est réduit d'une dimension ce qui correspond à supprimer la  $j^{\text{ième}}$  ligne et la  $j^{\text{ième}}$  colonne de l'approximation du Hessien  $G$  pour obtenir  $\hat{G}$

Soit  $G = L D L^T$  alors il peut être montré que

$$\hat{G} = \bar{L} D \bar{L}^T$$

Où  $\bar{L}$  est une  $(n-1) \times n$  matrice obtenue en supprimant la  $j^{\text{ième}}$  ligne de  $L$ .

Donc  $\bar{L}$  à la forme suivante :

$$\bar{L} = \begin{bmatrix} \begin{array}{c|c} L_1 & \\ \hline L_2 & \end{array} \\ \hline \end{bmatrix} = \begin{bmatrix} \begin{array}{c|c|c} \bar{L}_1 & I & \bar{L}_2 \\ \hline \end{array} \end{bmatrix}$$

où

$$L = \begin{bmatrix} \begin{array}{c|c} L_1 & \\ \hline * & * \\ \hline L_2 & \end{array} \\ \hline \end{bmatrix} \quad \leftarrow j^{\text{ième}} \text{ ligne}$$

et par conséquent la factorisation de Cholesky de  $\hat{G}$  est donnée par :

$$\hat{G} = \hat{L} \hat{D} \hat{L}^T$$

où

$$\hat{L} = \begin{bmatrix} \begin{array}{c|c} \bar{L}_1 & \\ \hline \bar{L}_2 & \end{array} \\ \hline \end{bmatrix} \quad \text{et} \quad \hat{D} = \begin{bmatrix} D_1 & \\ & \hat{D}_2 \end{bmatrix} \quad \text{avec} \quad D = \begin{bmatrix} D_1 & & \\ & d & \\ & & D_2 \end{bmatrix}$$

et  $\hat{L}_2$  et  $\hat{D}_2$  sont fournies par :

$$\hat{L}_2 \hat{D}_2 \hat{L}_2^T = \bar{L}_2 D_2 \bar{L}_2^T + d \mathbf{1}\mathbf{1}^T \quad (6)$$

Les matrices  $\hat{L}_2$  et  $\hat{D}_2$  peuvent être déterminés en utilisant le premier algorithme décrit dans la section 3.1.0.3.

### 3.2.1. Structure d'implémentation et résolution informatique du problème selon NAG.

#### 3.2.1.1. Paramètres principaux de la routine de NAG

##### A) Paramètres d'entrée.

N (entier) doit spécifier le nombre de variables indépendantes du problème.

INTYPE - ( entier )

INTYPE = 0 si aucune valeur n'est attribuée à F,G, HESL,HESD et à ISTATE.

INTYPE = 1 si les paramètres F et G contiennent respectivement la valeur de F (x) et de ses dérivées premières au point de départ et si HESD possède une estimation de la diagonale du hessien et HESL a été initialisé à la matrice nulle.

INTYPE = 2 si les paramètres F et G contiennent respectivement la valeur de F (x) et de ses dérivées premières au point de départ. De plus, les N éléments de ISTATE doivent indiquer si la variable correspondante est fixée sur une borne, ou sa position dans l'ensemble des variables libres et HESD et HESL doivent contenir les facteurs de Cholesky de la matrice hessienne dans le sous espace des variables libres.

INTYPE = 3 si aucune valeur n'est assignée à F et G  
mais que HESL, HESD et ISTATE sont comme  
pour INTYPE = 2.

MAXCAL ( Entier ) nombre maximal d'itérations.

ETA ( Réel )  $0.0. \leq \text{Eta} \leq 1$  doit spécifier la précision  
avec laquelle la recherche linéaire doit  
être effectuée

XTOL (Réel) spécifie la précision avec laquelle l'estima-  
tion du minimum doit être trouvée.

STEMX (Réel) fournit une estimation de la distance euclidien-  
ne entre la solution et le point de départ.

FEST (Réel) doit spécifier une estimation sur la valeur  
de la fonction au minimum.

IBOUND (Entier)

IBOUND = 0 si toutes les variables sont bornées et  
que l'utilisateur fournit leur valeur.

IBOUND = 1 si le problème est sans contrainte.

IBOUND = 2 si toutes les bornes sont du type  $0 \leq X_j$

IBOUND = 3 si toutes les variables sont bornées avec

$$L_1' = L_2' \dots = L_n' \text{ et } U_1' = U_2' \dots = U_n'$$

BL ( vecteur de réel )

BL ( j ) contient la  $j^{\text{ième}}$  borne inférieure.

si IBOUND = 1 ou 2 la routine initialise le vecteur.

si IBOUND = 3 alors  $BL(1) = L_1$  et la routine initialise  
les (n-1) autres bornes.

BU ( vecteur de Réels )

BU ( j ) contient la  $j^{\text{ième}}$  borne supérieure.

si IBOUND = 1 ou 2 la routine initialise le vecteur.  
 si IBOUND = 3 alors BU (1) = U<sup>1</sup> et la routine initialise  
 les (n-1) autres bornes.  
 LOCSCH ( logique ) doit spécifier si une recherche liné-  
 aire doit être effectuée quand un point vérifiant  
 les critères d'arrêt a été trouvé. Ceci permet  
 d'éviter les points de selle.

#### B) Paramètres d'entrée/sortie.

$\bar{X}$  ( vecteur de réels ) : estimation du minimum.

$\bar{F}$  ( Réel ) : valeur de  $F(x)$  au point  $X$ .

$\bar{G}$  ( vecteur de réels ) : valeur des dérivées premières.

de  $F(x)$  au point  $X$

$\overline{HESL}$  ( vecteur de réels de dimensions  $N \times (N-1)/2$  ) et  $\overline{HESD}$

( vecteur de réel ) sont utilisés pour stocker les

facteurs de Cholesky de l'approximation du hessien.

$\overline{ISTATE}$  ( vecteur d'entier )

$ISTATE(j) = -1$  si la variable est fixée sur

la borne supérieure.

$= -2$  si la variable est fixée sur la

borne inférieure.

$= -3$  si la variable est constante.

$= K$  si la variable est libre où

$K$  est sa position dans la séquence des

variables libres.



### 3.2.1.2. Initialisation.

Etape 1 : initialisation des contraintes de bornes  
 si IBOUND = 1, le problème est sans contrainte.

Toutes les bornes inférieures sont fixées à  $-10^6$  et les bornes supérieures à  $10^6$ .

si IBOUND = 2, les contraintes sont du type  
 $0 \leq x_j$

Les éléments du vecteur des bornes inférieures sont initialisés à zéro et ceux du vecteur des bornes supérieures au plus grand nombre positif représentable en machine.

Si IBOUND = 3, les bornes inférieures sont toutes égales entre elles et les bornes supérieures possèdent la même propriété.

Les éléments du vecteur des bornes inférieures et du vecteur des bornes supérieures sont tous initialisés à BL (1) et BU (1) respectivement.

Etape 2 : Si INTYPE  $\geq$  1

On teste si le point initial X est admissible et si ISTATE a été correctement rempli ,

Quand la valeur de la fonction et du gradient et (ou) les facteurs de Cholesky du hessien projeté évalué au point initial ont été



fournis par l'utilisateur (  $INTYPE > 0$  ), nous vérifions si les variables sont admissibles. Si une des variables viole une borne,  $INTYPE$  est mis à  $-1$  pour signaler que les valeurs fournies doivent être recalculées.

Si toutes les variables sont admissibles ( c'est à dire le point de départ est admissible ) et si le vecteur des états des variables a été communiqué par l'utilisateur (  $INTYPE = 2$  ou  $3$  ) on vérifie si chaque variable, fixée  $X_j$ , se trouve réellement à proximité d'une borne. Si une erreur est rencontrée, nous donnons la valeur  $-3$  à  $INTYPE$  si F et G doivent être recalculées, et la valeur  $-2$  dans le cas contraire.

Si l'état des variables a été correctement donné nous définissons une variable  $NFREE$  contenant le nombre de variable libres et une variable  $NEQUAL$  indiquant le nombre de variables constantes.

Par conséquent nous effectuons les tests suivants :

teste si  $x$  est admissible.

Soit  $Tol = \sqrt{\epsilon}$  ou  $\epsilon$  est le plus petit réel positif tel que  $1 + \epsilon > 1$ .

#1 pour  $j = 1, \dots, n$

si  $BL_j - x_j \leq Tol \times (|BL_j| + 1)$  alors

si  $x_j - BU_j \leq Tol \times (|BL_j| + 1)$  alors

aller en # 2.

sinon  $INTYPE = -1$  aller en # 4

sinon  $INTYPE = -1$  aller en # 4

#2 continuer

teste si ISTATE a été correctement rempli

#3 si  $INTYPE \geq 2$

pour  $j = 1, \dots, n$

si  $BU_j - BL_j \leq Tol \times (|BL_j| + 1)$

alors si  $x_j$  est définie comme variable fixée nous passons à la variable suivante en augmentant NEQUAL d'une unité et en mettant ISTATE (j) à -3

sinon  $INTYPE = -INTYPE$

si  $INTYPE = -3$  aller # 4

sinon  $ISTATE = -3$

sinon

si  $ISTATE(j) > 0$  on augmente le compteur de variable libre d'une unité et nous envisageons la variable suivante.  
si  $ISTATE(j) = -1$  on vérifie si

$$x_j \in [BL_j, BL_j + Tol (|BL_j| + 1)]$$

*si oui nous pouvons passer à la variable suivante.*

*sinon INTYPE = -3, aller en #4*

*si ISTATE (j) = - 1*

*si  $x_j \in [BU_j - Tol (|BU_j| + 1), BU_j]$*

*alors recommencer les tests pour la variable suivante*

*sinon INTYPE = - 3 aller en #4*

**#4 FIN**

Etape 3. si INTYPE = - 2

Initialiser les facteurs de Cholesky de l'approximation du hessien à ceux de la matrice identité et le nombre du conditionnement de la matrice hessienne à l'unité.

Etape 4 : Si INTYPE  $\leq$  - 2 alors INTYPE = INTYPE + 3

car ainsi si INTYPE était égale à - 3, il devient égale à 0 ce qui indique que tout doit encore être calculé et si INTYPE valait - 2 il devient + 1 ce qui signale que le vecteur ISTATE doit être évalué.

Etape 5 : Initialisation ou correction du vecteur des états des variables et recherche d'un point admissible si nécessaire.

Si  $INTYPE \leq 1$ ; c'est à dire si le vecteur des états des variables ISTATE n'a pas été communiqué ou s'il a été incorrectement initialisé par l'utilisateur ou si le point de départ n'est pas admissible, on vérifie la valeur initiale de X. Chaque variable très proche ou au-delà d'une borne est fixée sur cette borne et l'élément correspondant de ISTATE est mis à - 2 pour une borne inférieure, à -1 pour une borne supérieure et - 3 pour une variable constante.

Pour chaque variable libre  $X_j$ , ISTATE (j) contient sa position dans les permutations des variables libres. Le nombre de variables libres et le nombre de variables constantes sont évalués.

$NFREE = 0$ ,  $NEQUAL = 0$ ,  $Tol = \sqrt{\epsilon ps}$  où  $\epsilon ps =$  le plus petit nombre positif tel que  $1 + \epsilon ps > 1$

Pour  $j = 1, \dots, N$

- #1 si  $X_j - BL_j < BL_j + Tol (|BL_j| + 1)$  alors  
 $X_j = BL_j$ ; ISTATE (j) = - 2 ; aller en # 3
- #2 si  $BU_j - X_j < BU_j + Tol (|BL_j| + 1)$  alors  
 $X_j = BU_j$ ; ISTATE (j) = - 1 aller en #3  
 sinon aller en # 4
- #3 si  $BU_j = BL_j < Tol (|BU_j| + 1)$   
 $ISTATE (j) = - 3$   
 $NEQUAL = NEQUAL + 1$
- #4  $NFREE = NFREE + 1$

Etape 6 : Calculer la norme euclidienne du point de départ .

Etape 7 : Si  $INTYPE = 1$  ou  $INTYPE = 2$  alors

- si le nombre de variables constantes est égal à  $N$  nous allons à l'étape 33 du corps de boucle car la minimisation n'a aucun sens.
- s'il n'y a pas de variable libre nous allons à l'étape 23 du corps de boucle pour vérifier les multiplicateurs de Lagrange associés aux variables fixées non-constantes.

Si  $INTYPE = 2$  aller à l'étape 9.

Sinon l'utilisateur a fourni la valeur du gradient de la fonction au point  $X$  et une approximation de la diagonale du hessien ou nous sommes passés par l'étape 3.

Dans ce cas nous stockons les éléments du gradient correspondant aux variables libres dans l'ordre des permutations des variables non-fixées dans un vecteur pour obtenir le gradient projeté  $\bar{g}$  nous réalisons la même opération pour les facteurs de Cholesky de l'approximation du hessien et calculons la norme euclidienne du gradient projeté.

Nous passons directement à l'étape 10.



Etape 8 : Si  $\text{INTYPE} \neq 1$  et  $\text{INTYPE} \neq 2$

- calculer la fonction et ses dérivées premières au point  $X$ .  $\text{NF}=1$  ( nombre d'évaluations de la fonction ).
- si le nombre de variables constantes est égal à  $N$  nous allons à l'étape 33 du corps de boucle car la minimisation n'a aucun sens.
- s'il n'y a pas de variable libre nous allons à l'étape 23 du corps de boucle pour évaluer les multiplicateurs de Lagrange associés aux variables fixées non-constantes.

Etape 9. Stocker les éléments du gradient correspondant aux variables libres dans l'ordre des permutations des variables non-fixées dans un vecteur pour obtenir le gradient projeté  $\bar{g}$ .

Etape 10 Si  $\text{INTYPE} \leq 0$  alors

Quand les facteurs de Cholesky de l'approximation du hessien dans l'espace réduit n'ont pas été spécifiées par l'utilisateur ils sont initialisés à ceux de la matrice identité de l'espace des variables libres et les éléments restant du HESL et HESD sont initialisés à zéro. Le nombre du conditionnement de la matrice est égal à un.

Etape 11. Calculer la borne supérieure sur le conditionnement du hessien.

$\text{BOUNDK} = 10^{-2} / ( \text{NFREE} )^{\frac{1}{2}} \times \text{Eps}$  où Eps est le plus petit nombre positif tel que  $1 + \text{Eps} > 1$

Etape 12. Calculer l'estimation du nombre COND du conditionnement de la matrice d'approximation du hessien si  $\text{INTYPE} \geq 1$ .

C'est à dire rechercher le minimum  $D_s$  et le maximum  $D_B$  de la diagonale de la matrice diagonale des facteurs de Cholesky de l'approximation du hessien.

Si  $D_s < \text{Eps}$  alors  $\text{COND} = \text{RMAX}$

sinon  $\text{COND} = D_B / D_s$

si  $\text{BOUNDK} \leq \text{COND}$  alors  $\text{COND} = \text{RMAX}$

où RMAX est le plus grand réel positif représentable en machine.

Etape 13. Si  $\|\bar{g}\| < 10^{-2} (1 + |F|)^2$  aller à l'étape 23.  
du corps de boucle sinon aller à l'étape 0  
du corps de boucle.



### 3.3.1.3. Corps de boucle.

Etape 0. Tester si le nombre d'évaluations de la fonction n'est pas supérieur au nombre maximal permis.  
 si  $NF \leq MAXCAL$  aller en 1.  
 sinon aller en 33.

A) Calcul de la direction de descente P.  
 ( Etapes 1 à 3 )

Etape 1. a) calculer la direction de recherche P dans l'espace réduit en résolvant le système.

$$L^{(k)} D^{(k)} L^{(k)T} P = \bar{g} \text{ où } \bar{g} \text{ est le gradient projeté.}$$

Par les relations données dans la section 3.1.1.1.

b) Calculer la norme P

c) calculer le produit scalaire de - P par  $\bar{g}$

Etape 2. Etendre la direction de recherche P dans l'espace à n dimensions pour obtenir la direction  $P^*$ , de la façon suivante :

Pour  $j = 1, \dots, n$

Si  $ISTATE(j) < 0$  alors  $P^*(j) = 0$

sinon  $P^*(j) = P(ISTATE(j))$

Etape 3. Calculer la norme  $\bar{g}$

B) Impression des résultats.

( Etapes 4 et 5 )

Etape 4. Calculer l'estimation COND du nombre du conditionnement de la matrice d'approximation du hessien de la manière suivante.

Rechercher le minimum  $D_s$  et le maximum  $D_B$  de la diagonale de la matrice diagonale des facteurs de Cholesky de l'approximation du hessien

Si  $D_s < \text{EPs}$  alors  $\text{COND} = \text{RMAX}$

sinon  $\text{COND} = D_B / D_s$

si  $\text{BOUNDK} = 10^{-2/\sqrt{\text{NFREE}}} \text{EPs} \leq \text{COND}$

alors  $\text{COND} = \text{RMAX}$

Où  $\text{EPs}$  est le plus petit réel positif tel

$1 + \text{EPs} > 1.$

$\text{RMAX}$  est le plus grand réel positif représentable en machine.

et  $\text{BOUNDK}$  est la borne supérieure permise sur le nombre de conditionnement de la matrice hessienne.

Etape 5. Impression des résultats.

C) Recherche linéaire.

( Etapes 6 à 12 )

Etape 6. Calculer le pas maximum pour la recherche linéaire.

$$\begin{aligned} \text{SPE} &= \text{STEPMX} / P \quad \text{si STEPMX} / p \text{ n'engendre} \\ &\quad \text{pas d'overflow} \\ &= \text{FLMAX} \quad \text{si } P = 0 \text{ ou si STEPMX} \\ &\quad \text{provoque un overflow} \end{aligned}$$

Avec FLMAX = 1'unité divisée par le plus petit nombre positif représentable en machine, tel que  $-R$ ,  $1/R$ ,  $-1/R$  peut être calculé sans créer d'overflow ou underflow

Etape 7. Modifier le pas maximum pour obtenir un pas maximum admissible.

$$\begin{aligned} \text{. Rechercher } \max &= \min \left\{ \text{SPE}, \alpha_i \quad i=1..N\text{FREE} \right\} \\ \text{où } \alpha_i &= \max \left\{ \frac{1-x_i}{P_i}, \frac{i-x_i}{P_i}; P_i \neq 0 \right\} \end{aligned}$$

- . Si  $\alpha_{\max}$  est négligeable, fixer la variable correspondante à  $\alpha_i = \alpha_{\max}$  sur sa borne.  
Une variable  $J_{\text{fix}}$  doit permettre de savoir si une variable a été fixée ou non, et désigner laquelle si nécessaire.

Par conséquent il faut effectuer les tests suivant

$$\alpha_{\max} = \text{SPE}; J_{\text{fix}} = 0; \epsilon_{\text{ps}} = \text{le plus petit réel positif tel que } 1 + \epsilon_{\text{ps}} > 1.$$

$$\text{Pour } j = 1, \dots, n$$

#1 si  $|P_j| < \epsilon \rho s^2$  aller en #4 car la variable correspondante est une variable fixée sur une borne.

#2 si  $P_j < 0$  nous devons considérer le pas en fonction de la borne inférieure associée à la  $j^{\text{ème}}$  variable.

$$\text{soit } X_{\text{dif}} = X_j - BL_j$$

$$\text{si } X_{\text{dif}} > \sqrt{\epsilon \rho s} (|BL_j| + 1)$$

$$\text{alors si } -X_{\text{dif}} / P_j < \alpha_{\text{max}}$$

$$\text{alors } \alpha_{\text{max}} = -X_{\text{dif}} / P_j$$

sinon aller en #4

sinon le pas étant négligeable on fixe la  $j^{\text{ème}}$  variable sur sa borne.

$$X_j = BL_j,$$

$$ISTATE(j) = -2,$$

$$J_{\text{fix}} = j, \text{ aller en } \#5$$

#3 si  $P_j > 0$  ( le pas est lié à la borne supérieure

$$\text{soit } X_{\text{dif}} = BU_j - X_j$$

$$\text{si } -X_{\text{dif}} > \sqrt{\epsilon \rho s} (|BU_j| + 1)$$

$$\text{alors si } X_{\text{dif}} / P_j < \alpha_{\text{max}} \text{ alors}$$

$$= \alpha_{\text{max}} = X_{\text{dif}} / P_j$$

sinon aller

en #4

$$\text{sinon } X_j = BU_j ; ISTATE(j) = -1$$

$$\text{et } J_{\text{fix}} = j,$$

aller en #5

#4 continuer.

#5  $NFREE = NFREE - 1$

Corriger le vecteur des états des variables

Pour  $j = 1, \dots, N$

si  $ISTATE(j) > J_{fix}$  alors  $ISTATE(j)$   
 $= ISTATE(j) - 1.$

Calculer la borne supérieure du nombre  
 de conditionnement de la matrice hessienne

$$BOUNDK = 10^{-2} / \sqrt{NFREE} \times \epsilon_{ps}$$

Etape 8. Si  $J_{fix} = 0$  aller à l'étape 11.

Sinon aller à l'étape 9.

Etape 9. Une nouvelle variable a été fixée.

a) Recalculer la fonction

$$NF = NF + 1$$

b) Modifier les facteurs de Cholesky du  
 hessien projeté de la façon suivante :

Soit  $j = J_{fix}$

Si  $j = NF + 1$  ou si  $NFREE = 0$  aller en(\*)

Sinon copier la  $j^{ième}$  colonne de la

matrice triangulaire inférieure

de Cholesky dans un vecteur auxil-  
 iaire.

supprimer la  $j^{ième}$  colonne de la  
 matrice triangulaire en compac-  
 tant la matrice.

- copier le  $j^{\text{ième}}$  élément de la diagonale de la matrice diagonale de Cholesky dans une variable auxiliaire.

- supprimer le  $j^{\text{ième}}$  élément de la diagonale de la matrice diagonale en la compactant.

- effectuer la correction de rang 1 décrite au sous-chapitre 3.2.0.3. en utilisant l'algorithme expliqué dans la section 3.1.1.2. et aller en (\*\*)

(\*\*) • annuler le  $(\text{NFREE} + 1)^{\text{ième}}$  élément de la diagonale de la matrice diagonale et si  $\text{NFREE} \neq 0$  annuler la  $(\text{NFREE} + 1)^{\text{ième}}$  ligne de la matrice triangulaire inférieure.

c) Réduire le gradient projeté au nouvel espace de variables libres.

Pour  $j = 1, \dots, N$

si  $\text{ISTATE}(j) > 0$  alors  $\bar{g}$  ( $\text{ISTATE}(j) = g(j)$ )

d). Si  $\text{NFREE} = 0$  aller à l'étape 23 pour évaluer les multiplicateurs de Lagrange.

- Calculer  $\|\bar{g}\|$



Etape 10. Retour à l'étape 0.

Etape 11. Calculer la longueur du pas initial le long de la direction P pour la recherche linéaire. Prendre le pas initial  $\alpha$  tel que

$$|FEST - F| \geq -\alpha \mu g^T P^*$$

où F est la valeur de la fonction en X

FEST est une estimation de la valeur de la fonction au minimum et  $\mu = \frac{1}{2}$

le problème est résolu par

$$\text{Soit } D = 2 \times |FEST - F|$$

$$\text{Si } D \leq -\langle g, p^* \rangle \text{ et } D > \text{Eps alors } \alpha = -D / \langle g, p^* \rangle$$

$$\text{sinon } \alpha = 1$$

$$\text{Si } \alpha > \alpha_{\max} \text{ alors } \alpha = \alpha_{\max}$$

Etape 12. Effectuer la recherche linéaire pour obtenir  $x^{(k+1)}$ ,  $F^{(k+1)}$  et  $g^{(k+1)}$

D) La recherche linéaire échoue.

( Etapes 13 et 14 )

Etape 13. Si la recherche linéaire échoue aller à l'étape 14  
sinon passer à l'étape 15.

Etape 14. Quand la recherche linéaire n'a pas trouvé un meilleur point c'est à dire un point tel que

$$|g(x + \alpha^{(k)} p^*)^T p^*| < -\eta g^T p^* \text{ avec } \eta \in (0, 1]$$

et

$$F^{(k)} - F^{(k+1)} > -\alpha^{(k)} \mu g^T p^* \text{ avec } \mu \in (0, 1]$$

alors si le gradient est déjà petit ( $\|g\| < \epsilon \rho^{2/3} (1 + \|F\|^2)$ )

ou que la borne supérieure sur le nombre du conditionnement du hessien a déjà été réduite ( $\text{BOUNDK} \leq 100$ ), alors si  $\text{BOUNDK} > 1$  nous réinitialisons l'approximation du hessien à la matrice identité et nous mettons  $\text{BOUNDK} = 1$  et retournons à l'étape 0.

sinon nous vérifions les multiplicateurs de Lagrange de chaque variable fixée non-constante si  $(N - \text{NFREE}) > \text{NEQUAL}$

( nous allons par conséquent à l'étape 23. ) et

si  $(N - \text{NFREE}) \leq \text{NEQUAL}$

nous allons à l'étape 32. pour effectuer une recherche linéaire spéciale.

sinon

nous réduisons la borne supérieure sur le nombre du conditionnement de la matrice hessienne et nous modifions le conditionnement de la matrice diagonale des facteurs de Cholesky de l'approximation du hessien.

$$BOUNDK = (BOUNDK)^{\frac{1}{2}}$$

POUR  $i = 1 \dots NFREE$

$$D_i = (D_i)^{\frac{1}{2}}$$

et nous retournons à l'étape 0

E) Correction de rang deux.

(Etapes 15 à 22)

Etape 15. Lorsque la recherche linéaire s'est terminée avec succès, nous testons si une variable quelconque a atteint l'une de ses bornes. Dans ce cas, la variable est fixée sur la borne, une variable auxiliaire  $J_{fix}$  reçoit l'indice de la variable dans les permutations des variables libres. Le nombre des variables libres est diminué d'une unité. La borne supérieure sur le nombre du conditionnement du hessien projeté est recalculée et la norme du nouveau point ainsi que la valeur de la fonction en ce point sont évaluées

Recalcule. = false;  $J_{fix} = 0$

Pour  $j = 1 \dots n$

#1 si  $ISTATE(j) < 0$  aller en #3

#2 soit  $X_{diff} = X_i - BL_j$

si  $P_j < 0$  alors

Si  $X_{dif} > \sqrt{\epsilon_{ps}} (|BL_j| + 1)$  aller en #3  
 Si  $X_{dif} \geq 10 \sqrt{\epsilon_{ps}} (|BL_j| + 1)$  recalcule = true  
 $X_j = BL_j$   
 $J_{fix} = ISTATE(j); ISTATE(j) = -2$  aller #4  
sinon  
 Soit  $X_{dif} = BU_j - X_j$   
 Si  $X_{dif} > \sqrt{\epsilon_{ps}} (|BU_j| + 1)$  aller en #3  
 Si  $X_{dif} \geq 10 \sqrt{\epsilon_{ps}} (|BU_j| + 1)$  recalcule = true  
 $X_j = BU_j, J_{fix} = ISTATE(j),$   
 $Istate(j) = -1,$  aller en #4  
 #3 continuer  
 #4 si  $J_{fix} > 0$  alors  
 $NFREE = NFREE - 1$   
 $BOUNDK = 10^{-2} / \sqrt{NFREE} \epsilon_{ps}$   
 Pour  $j = 1..N$   
 si  $ISTATE(j) > J_{fix}$   $ISTATE(j) =$   
 $ISTATE(j) - 1$   
 #5 calculer  $\|X\|$   
 #6 si RECALCULE = true, évaluer  $F(x)$  et  $NF = NF + 1$

Etape 16. Si une variable a été fixée sur une borne c'est  
 -à-dire  $J_{fix} \neq 0$  nous réduisons les facteurs  
 de Cholesky du hessien projeté en utilisant le  
 même procédé qu'à l'étape 9.

Etape 17. Si le nombre de variables libres est nul nous passons à l'étape 23 pour évaluer les multiplicateurs de Lagrange des variables fixées non-constantes.

Etape 18. a) nous réduisons le nouveau gradient  $g^{(k+1)}$  dans l'espace des variables libres pour obtenir le gradient projeté et nous calculons sa norme ( voir 9.C)  
 b) si  $J_{fix} > 0$ , une variable a été fixée, nous supprimons le  $J_{fix}^{ième}$  élément du gradient  $\bar{g}$  et de la direction  $p$  et calculons le produit scalaire de  $\bar{g}$  par  $P$

$$P^T \bar{g} = P^T \bar{g} - P_{J_{fix}} g_{J_{fix}} \quad \left( \text{car } P^T \bar{g} \text{ a déjà été calculé} \right)$$

Pour  $j = J_{fix}, \dots, N_{free}$

$$P(j) = P(j+1)$$

$$\bar{g}(j) = \bar{g}(j+1)$$

c) nous calculons -  $\langle \bar{g}^{(k+1)}, P \rangle$  et  $\|\bar{g}^{(k+1)}\|$

d) si  $\|\bar{g}^{(k+1)}\| \geq 10x \text{ XTOL}$  aller à l'étape 19  
 sinon aller à l'étape 20 avec une variable  
 $Iflag = 0$

Etape 19. Effectuer la correction de rang deux

Soit  $\alpha$ , le pas obtenu par la recherche linéaire  
 et Soit  $TEST = \alpha ( \langle P, \bar{g}^{(k+1)} \rangle - \langle P, \bar{g} \rangle )$

Si  $TEST > 0$  alors nous avons l'assurance que

$B^{(k+1)}$  est définie positive car :

$$B^{(k+1)} s^{(k)} = y^{(k)}$$

$$s^{(k)T} B^{(k+1)} s^{(k)} = s^{(k)T} y^{(k)} = Test > 0$$

Si  $TEST > 0$  alors

1) si  $TEST \geq 1$  aller en 2).

si  $TEST \times RMAX \leq 1$  aller en 6)

sinon aller en 2)

2) effectuer la correction de rang un  
positive par l'algorithme 1 de sec-  
tion 3.1.1.2.

3) Soit  $TEST = |\langle P, \bar{g} \rangle|$

si  $TEST \geq 1$  aller en 4)

si  $TEST \times RMAX \leq 1$  aller en 6)

sinon aller en 4)

4) Effectuer la correction de rang un  
négative par l'algorithme 2 de la  
section 3.1.1.2.

5)  $IFLAG = 1$  et aller à l'étape 20.

6)  $IFLAG = -1$  et aller à l'étape 20.

Sinon  $IFLAG = 0$



Etape 20.

Si IFLAG = - 1 aller à l'étape 23.  
sinon aller à l'étape 21.

Etape 21.

Si IFLAG > 0 nous testons l'approximation du nombre de conditionnement de la matrice.

Si le rapport du plus grand élément de la diagonale sur le plus petit de la diagonale de la matrice diagonale de Cholesky est plus grand que BOUNDK, la borne supérieure sur le nombre de conditionnement de la matrice, alors chaque élément de la diagonale de la matrice diagonale des facteurs de Cholesky du hessien est élevé à une puissance W qui est telle que la nouvelle estimation du nombre du conditionnement du hessien soit égale à BOUNDK.

W est donc égale à  $\log(\text{BOUNDK})/\log(\text{COND})$  où  $\text{COND} = \frac{\text{le plus grand élément de la diagonale}}{\text{le plus petit élément de la diagonale}}$ .

F) Test pour la convergence.

Etape 22. Si  $\|\bar{g}^{(k+1)}\| \geq 10^{-2} (1 + |F^{(k+1)}|)$  et

$$|F^{(k)} - F^{(k+1)}| > 10^{-1} (1 + |F^{(k)}|)$$

aller à l'étape 0 avec  $\bar{g} = \bar{g}^{(k+1)}$

Sinon

aller à l'étape 23.

G) Evaluation des multiplicateurs de Lagrange.  
( Etapes 23 et 24 ).

Etape 23. : INEGLM = 0

Si ( N - NFREE ) > NEQUAL nous calculons les multiplicateurs de Lagrange de chaque variable fixée  $X(j)$  non-constante. S'il n'est pas indiscutablement positif ou s'il est négatif, nous mettons ISTATE (j) de - 1 à - 4 ou de - 2 à -5. Quand tous les multiplicateurs ont été calculés, une variable INEGLM est mise à zéro si l'ensemble des multiplicateurs négatifs est vide, sinon une autre variable RNEGLM contiendra la valeur du multiplicateur le plus négatif et INEGLM son indice tandis qu'une variable NUMEG contiendra le nombre de multiplicateurs négatifs ou non-indiscutablement positifs. Dans le deuxième cas nous vérifions la dimension relative du multiplicateur le plus négatif et si elle est petite, INEGLM est remise à zéro pour indiquer qu'aucune variable ne doit être relâchée.

Si INEGLM > 0, nous remettons le vecteur des états des variables à sa valeur normale

$RNEG\text{LM} = (\epsilon ps)^{\frac{1}{2}}$

$INEG\text{LM} = 0$

$NUMEG = 0$

Pour  $j = 1..n$

#1 si  $ISTATE(j) \geq 0$  ou  $= -3$ , aller en #3

#2 si  $ISTATE(j) = -1$  (variable fixée sur une  
borne supérieure)

alors  $\lambda_j = -g_j^{(k+1)}$

sinon  $\lambda_j = g_j^{(k+1)}$

si  $\lambda_j > (\epsilon ps)^{\frac{1}{2}}$  aller en #3

$ISTATE(j) = ISTATE(j) - 3$

$NUMEG = NUMEG + 1$

si  $\lambda_j > RNEG\text{LM}$ , aller en #3

$RNEG\text{LM} = \lambda_j$

$INEG\text{LM} = j$

#3 continuer

#4 si  $INEG\text{LM} > 0$

si  $NFREE = 0$  alors si  $|\lambda_s| \leq \sqrt{\epsilon ps}$  alors  $INEG\text{LM} = 0$

sinon si  $\|\lambda\| / (NFREE)^{\frac{1}{2}} > \lambda_s$ ,  $INEG\text{LM} = 0$

où  $\lambda_s = \min_i \left\{ \lambda_i \right\}$

#5 si  $INEG\text{LM} > 0$

Remettre  $ISTATE$  à sa valeur normale.

Etape 24.

Si  $INEG\text{LM} > 0$  aller à l'étape 31.

sinon aller à l'étape 25.

H) Aucune contrainte ne sort de l'ensemble des contraintes actives.

Etape 25. Test des critères d'arrêt.

Si (NUMEG  $\neq$  0) et (NWHY=3 ou NFREE= 0) aller à l'étape 27.

Si (NFREE = 0) aller à l'étape 33.

Si (NWHY = 3) aller à l'étape 32.

Si (  $\|\bar{g}^{(k+1)}\| < 10^{-4} \varepsilon_{ps}(1+\bar{F}^{(k+1)})^2$  ) aller à l'étape 26'

Si (NITER = 0) aller à l'étape 26.

(a) Si  $\alpha \|P\| < (XTOL + \sqrt{\varepsilon_{ps}})(1+\|X\|)$  et  $|\bar{F}^{(k+1)} - \bar{F}^{(k)}| < (XTOL^2 + \varepsilon_{ps})(1+\bar{F}^{(k+1)})$  et  $\|g^{(k+1)}\| < (XTOL^2 + \varepsilon_{ps}) * (1+\bar{F}^{(k+1)})^2$  aller à l'étape 26'.

Etape 26. Si (NUMEG > 0) nous remettons sa valeur normale, au vecteur des états des variables ISTATE et retournons en 0.

Etape 26' Si NUMEG  $\neq$  0 aller à l'étape 27

Si l'utilisateur désire une dernière recherche linéaire c'est à dire LOCSCH= True aller à l'étape 32 sinon aller à l'étape 33.

Etape 27. Quand tous les multiplicateurs de Lagrange non-positifs associés aux variables fixées non-constantes sont proches de zéro et que le critère de convergence (a) est vérifié ou qu'il n'est pas possible de continuer la minimisation dans le sous-espace actuel, nous effectuons une petite perturbation admissible pour chaque variable dont les multiplicateurs sont proches de zéro. Si une telle perturbation permet de trouver une valeur de la fonction inférieure à l'ancienne, alors INEGLM retient l'indice de la variable perturbée; autrement nous donnons à INEGLM la valeur nulle, et nous remettons le vecteur des états des variables à sa valeur normale.

$$INEGLM = 0$$

#1 Pour  $j = 1, \dots, n$

$$\text{si } ISTATE(j) < -3$$

$$= 1/4 \text{ Eps } (1 + |x_j|)$$

$$\text{si } ISTATE(j) = -4, \quad \alpha = -\alpha$$

$$X_j = X_j + \alpha$$

$$\text{calculer } F(x); NF = NF + 1$$

$$\text{si } F(x) < F^{(k+1)}(x), \text{ INEGLM} = j \text{ et aller en \# 2}$$

$$X(j) = x(j) - \alpha$$

#2 Remettre ISTATE à sa valeur normale.

Etape 28. Si  $INEGLM = 0$  aller à l'étape 33.

Etape 29. Calculer la norme de  $X$

Etape 30. Le point perturbé devient le nouveau point de départ pour l'itération suivante. Nous réévaluons la fonction et son gradient en ce point et  
 $NF = NF + 1$

I) Une contrainte doit être relâchée.

Etape 31. Nous devons relâcher la variable  $X_{ineglm}$ , nous plaçons cette variable à la fin des permutations des variables libres et ajoutons une ligne et une colonne de la matrice identité, à la matrice d'approximation du hessien comme nous l'avons expliqué à la section 3.2.0.2. nous augmentons le gradient projeté et calculons sa norme ainsi que la borne supérieure sur le nombre du conditionnement du hessien. Si c'est nécessaire nous modifions les éléments de la diagonale de la matrice diagonale des facteurs de Cholesky de l'approximation du hessien de la même manière qu'à l'étape 21 et retournons à l'étape 0.



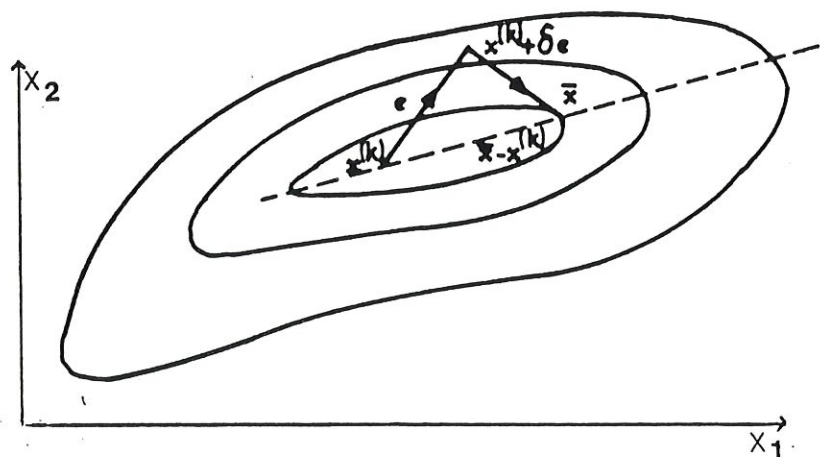
J) Recherche linéaire spéciale.

Etape 32. Nous imprimons les résultats et effectuons une recherche linéaire spéciale de la façon suivante :

Une première recherche linéaire est effectuée du point  $X + \delta e$ , où  $e = \sum_i e_i$  et  $\delta$  est un petit scalaire positif.

le long de la direction P orthogonal à e. Une seconde recherche linéaire est réalisée le long de la ligne reliant  $\bar{X}$  le minimum trouvé par la première recherche linéaire, et X

La figure suivante est l'interprétation géométrique de ces recherches linéaires.



#1. Calculer la direction de recherche P de la façon suivante

$$P_j = \begin{cases} 0 & \text{si la } j^{\text{ième}} \text{ variable est fixée sur une borne} \\ -1 & \text{si } x_j - BL_j > BU(j) - x_j \\ 1 & \text{sinon} \end{cases}$$

• Calculer le pas maximum admissible dans la direction positive.

$$\alpha_{\text{pos}} = \min_j \left\{ \alpha_j \right\}$$

$$\text{où } \alpha_j = \begin{cases} x_j - BL_j, & \text{si } P_j = -1 \\ BU_j - x_j, & \text{si } P_j = 1 \end{cases}$$

. Dans le cas spécial où il n'y a qu'une variable libre, calculer aussi le pas maximum admissible  $\alpha_{\text{neg}}$ , dans la direction négative.

$$\alpha_{\text{neg}} = \begin{cases} -(X_j - BL_j), & \text{si } P_j = 1 \\ -(BU_j - X_j), & \text{si } P_j = -1 \end{cases}$$

. Si le pas maximal admissible dans la direction positive est trop petit et que le nombre de variables libres est supérieur à un nous allons en E) avec un paramètre IFAIL = 1 pour indiquer que la recherche linéaire est inutile.

si  $\alpha_{\text{pos}} \geq \text{Eps}^{\frac{1}{2}}$  ( où Eps est + petit nombre positif tel que  $1 + \text{Eps} > 1$  )

alors aller en #B

sinon si NFREE > 1 aller en #E

sinon aller en #A

#A . si  $\alpha_{\text{neg}} > -\text{Eps}^{\frac{1}{2}}$  aller en #E ( pas dans la direction négative est trop petit )

sinon poser  $\alpha = \alpha_{\text{neg}}$

#B

#B1 , calculer  $Y = X + \alpha P$   
calculer  $F(y)$   
 $NF = NF + 1$

#B2 .si  $|F(Y) - F(x)| \geq \text{Eps} (1 + |F(x)|)$  aller en # C  
si le test est vérifié et si  $F(y) < F(x)$  le point y sera utilisé pour effectuer les deux recherches linéaires expliquées ci-dessus.  
Par contre si  $F(y) > F(x)$  nous n'effectuerons pas les recherches linéaires.

#B3 .si  $\alpha < 0$  , aller en # E avec  $\text{IFAIL} = 1$   
 $\alpha' = 5 \alpha$   
.ssi  $\alpha \leq \alpha_{\text{pos}}$  , aller en #B1;  
On augmente le pas  $\alpha$  jusqu'au moment où le test de # B2 est vérifié ou que le pas dépasse le pas maximum admissible.

#B4 .si  $\text{NFREE} = 1$  aller en # A.  
Dans le cas spécial où il n'y a qu'une variable libre si nous ne trouvons pas un Y tel que  $F(y) < F(x)$  dans la direction positive, nous réalisons une dernière tentative dans la direction négative.

#C .si  $\text{NFREE} = 1, \alpha > 0$  et  $F(y) > F(x)$  aller en # A

#D .Si  $NFREE > 1$  ou  $F(y) < F(x)$  alors  $IFAIL = 0$

Pour indiquer que les recherches linéaires  
peuvent être effectuées

#E .FIN

#2 .Si  $IFAIL \neq 0$  ou si  $NF > MAXCAL$  aller à l'étape  
33 du corps de boucle.

Sinon si  $NFREE = 1$  aller en # 3

sinon aller en # 4

#3 .Y devient le nouveau point de départ pour l'ité-  
ration suivante.

Nous retournons par conséquent à l'étape 0 du  
corps de boucle après avoir évalué la fonction  
et ses dérivées premières au nouveau point de  
départ.

#4 .Construire un vecteur orthogonal à  $(x-y)$  au point  
y en mettant chacun de ses éléments non-nuls al-  
ternativement à  $P(j)$  et  $-P(j)$ . Si  $NFREE$  est  
impaire mettre les deux derniers éléments du  
vecteur orthogonal à  $(x-y)$  de manière à ce que  
leurs modules diffèrent mais que leur norme  
reste égale à l'unité.

. Rechercher les longueurs des pas maximums admissibles

$\alpha_{\text{pos}}$  et  $\alpha_{\text{neg}}$  respectivement dans les directions positives et négatives.

$$\alpha_{\text{pos}} = \min_j (\alpha_j)$$

$$\alpha_j = \begin{cases} (BU_j - Y_j) / P_j & \text{si } P_j > 0 \\ (X_j - BL_j) / P_j & \text{si } P_j < 0 \end{cases}$$

$$\alpha_{\text{neg}} = \min_j (\alpha_j)$$

$$\alpha_j = \begin{cases} (BU_j - Y_j) / P_j & \text{si } P_j < 0 \\ (X_j - BL_j) / P_j & \text{si } P_j > 0 \end{cases}$$

. Calculer la direction de recherche P admissible, multiple d'un scalaire  $\mu$  significatif, du vecteur orthogonal.

. Si  $\alpha_{\text{pos}} > \alpha_{\text{neg}}$  nous envisagerons d'abord la recherche linéaire dans la direction positive.

. Si  $\alpha_{\text{pos}} < \text{Eps}$  où Eps est le plus petit nombre réel positif tel que  $1 + \text{Eps} > 1$

alors nous allons en #5 avec IFAIL = 1

. Si  $\alpha_{\text{pos}} < \alpha$  où  $\alpha$  a été obtenu en #1

alors nous posons  $\alpha = \alpha_{\text{neg}}$

$$\mu = \alpha$$

Nous recalculons les longueurs des pas maximums admissibles respectivement dans la direction négative et positive.

Si  $\alpha > 1$  alors  $\alpha_{\text{pos}} = \alpha_{\text{pos}} / \alpha$   
 $\alpha_{\text{neg}} = \alpha_{\text{neg}} / \alpha$

Sinon

si  $\alpha_{\text{pos}} < \alpha \times \text{RMAX}$  où RMAX est le plus grand réel positif tel que RMAX et - RMAX  
 alors  $\alpha_{\text{pos}} = \alpha_{\text{pos}} / \alpha$  peuvent être représentables en machine.

si  $\alpha_{\text{neg}} < \alpha \times \text{RMAX}$   
 alors  $\alpha_{\text{neg}} = \alpha_{\text{neg}} / \alpha$

Sinon dans ce cas la recherche linéaire dans la direction négative sera d'abord envisagée.

si  $\alpha_{\text{neg}} < \text{Eps}$  alors nous allons à # 5 avec  
 IFAIL = 1

si  $\alpha_{\text{neg}} < \alpha$  alors poser  $\alpha = \alpha_{\text{neg}}$   
 $\mu = -\alpha$

si  $\alpha > 1$  alors nous permutons les valeurs du pas maximum admissible car nous avons changé le signe de P

$\alpha_n = \alpha_{\text{neg}}$   
 $\alpha_{\text{neg}} = \alpha_{\text{pos}} / \alpha$   
 $\alpha_{\text{pos}} = \alpha_{\text{neg}} / \alpha$   
 sinon si  $\alpha_{\text{pos}} < \alpha \times \text{RMAX}$  =  $\alpha_{\text{neg}} = \alpha_{\text{pos}} / \alpha$   
si  $\alpha_n < \alpha \times \text{RMAX}$  =  $\alpha_{\text{pos}} = \alpha_n / \alpha$



.Poser  $P = \mu P$  et IFAIL = 0

- #5 .Si IFAIL n'est pas égale à zéro, les recherches linéaires ne sont pas effectuées car le pas maximal admissible est trop petit. Par conséquent dans ce cas, nous allons à l'étape 33. Par contre, si IFAIL = 0 nous allons en # 6 pour effectuer la première recherche linéaire.
- #6 .Calculer le produit scalaire du gradient de la fonction évalué en y par la direction de recherche P. S'il est négatif alors P est une direction de descente. S'il est positif nous passons  $P = -P$  et  $\alpha_{\text{pos}} = \alpha_{\text{neg}}$  et par conséquent nous réalisons la recherche linéaire dans la direction négative qui est une direction de descente.
- . Si  $\alpha_{\text{pos}} < \text{Eps}$  nous allons à l'étape 33 car le pas maximum admissible est trop petit.
- . Calculer la norme de P
- . Calculer le pas maximum  $\alpha_{\text{max}}$  comme à l'étape 6 du corps de boucle.
- . Si  $\alpha_{\text{max}} > \alpha_{\text{pos}}$  poser  $\alpha_{\text{max}} = \alpha_{\text{pos}}$
- . Calculer le pas initial pour la recherche linéaire comme à l'étape 11.

. EFFECTUER LA RECHERCHE LINEAIRE.

. Si le nombre maximal d'évaluations de la fonction permis est dépassé aller à l'étape 33.

Sinon aller en # 7.

#7 . Nous réalisons la deuxième recherche linéaire qui s'effectue le long de la direction  $X - z$  où  $z$  est le point obtenu par la première recherche linéaire.

. Si la recherche linéaire précédente nous a fourni un point  $z$  tel que  $F(z) > F(x)$  alors nous calculons les longueurs des pas maximums admissibles  $\alpha_{neg}$  et  $\alpha_{pos}$  le long de  $P = x - z$  à partir de  $x$  respectivement dans la direction négative et positive.

par contre, si la recherche linéaire nous a communiqué un point  $z$  tel que  $F(z) > F(x)$  alors nous calculons les longueurs des pas maximums admissibles  $\alpha_{neg}$  et  $\alpha_{pos}$  le long de  $P$  à partir du point  $z$  respectivement dans la direction négative et positive.

$$\alpha_{pos} = \min_j \{ \alpha_j \}$$

$$\text{où } \alpha_j = \begin{cases} (BU_j - W_j) / P_j & \text{si } P_j > 0 \\ -(W_j - BL_j) / P_j & \text{si } P_j < 0 \end{cases}$$

$$\alpha_{\text{neg}} = \min_j \alpha_j$$

$$\text{où } \alpha_j = \begin{cases} (BU_j - W_j) / P_j & \text{si } P_j < 0 \\ -(W_j - BL_j) / P_j & \text{si } P_j > 0 \end{cases}$$

où  $w = x$  ou  $z$  suivant les cas.

. Effectuer les différents pas de # 6 pour obtenir un nouveau point  $x^*$

#8 . Si la valeur de la fonction n'est pas diminuée passons à l'étape 33.

Dans le cas inverse, nous prenons le point  $x^*$  obtenu par la deuxième recherche linéaire comme nouveau point de départ, mais avant de retourner à l'étape 0 du corps de boucle nous devons tester si une variable a atteint une de ses bornes ce qui se réalise comme à l'étape 15.

Si une variable a été fixée, nous modifions les facteurs de Cholesky de l'approximation du hessien de manière identique qu'à l'étape 16 et nous allons en #9.

. Si aucune variable n'a été fixée nous allons en #10

#9 . Si le nombre de variables libres est nul aller en #10.

. Nous réduisons le gradient de la fonction évalué au point  $x^*$  pour obtenir le gradient projeté ( voir l'étape 9 c)

. Test d'arrêt.

si (  $|F(x) - F(x^*)| < (XTOL^2 + Eps)(1 + |F(x^*)|)$  )  
 et  $\left[ \|\bar{g}\| < (XTOL^2 + Eps^{2/3})(1 + |F(x^*)|)^2 \right.$   
 ou  $\left. \|\bar{g}\| < 10^{-4} Eps \right]$  alors aller en #10  
sinon aller à l'étape 0  
 du corps de boucle.

#10 .si le test de convergence est satisfait mais qu'une variable a été fixée nous vérifions les multiplicateurs de Lagrange associés aux variables fixées non-constantes de façon identique à l'étape 23, puis nous remettons le vecteur des états des variables à sa valeur initiale s'il a été modifié par l'exécution de l'étape 23.

Si tous les multiplicateurs sont positifs nous passons à l'étape 33 ( fin de l'algorithme ),  
sinon on recherche la contrainte dont le multiplicateur associé est le plus négatif et on modifie les facteurs de Cholesky de l'approximation du hessien comme à l'étape 31 avant de retourner à l'étape 0.

K) Fin de l'algorithme.

Etape 33.      Impression des résultats.

### 3.2.2. Implémentation graphique.

Le logiciel d'aide graphique à l'optimisation OPTIGRAPH développé ( voir chapitre 4 ) requiert dans le cas de minimisation avec contraintes de bornes utilisant la stratégie des contraintes actives, les informations suivantes :

- NINIT : la dimension de l'espace des variables.
- N : la dimension de l'espace réduit.
- g : le gradient projeté de la fonction au point courant.
- p : la direction de descente (  $P \in \mathbb{R}^n$  )
- H : le hessien ou une approximation du hessien au point courant. Il peut être communiqué sous deux formes, soit factorisé (Cholesky) soit non-factorisé.
- ISTATE: le vecteur des états des variables  

$$\text{ISTATE}(j) < 0 \quad \text{si la } j^{\text{ième}} \text{ variable est fixée sur une borne.}$$

$$\text{ISTATE}(j) = k \quad \text{si la } j^{\text{ième}} \text{ variable est libre où } k \text{ est sa position dans la séquence des variables libres.}$$

Les caractéristiques du logiciel sont identiques à celles développées dans la section 3.1.2.



3.2.3. Exemple.

Soit à maximiser la probabilité de choisir un des  $r$  meilleurs candidats suivant une certaine stratégie.

La fonction  $f_n$  représentant cette probabilité de succès est donné par

$$f_n = \frac{r}{n} \sum_{j=0}^{r-1} x \dots x_j \binom{n}{j} (1 - x_{j+1})^{n-j}$$

$$\sum_{k=1}^r \sum_{l=1}^r \sum_{j=\max(k,l)+1}^n x_1 \dots x_l \binom{j-2}{l-1} \frac{(1-x_1)^{j-1} - (1-x_{l+1})^{j-1}}{j-1}$$

où  $r$  est le nombre de candidats à choisir.

$n$  le nombre de candidats qui se présentent

et  $x_j \in (0,1)$   $j = 1..r$

Supposons  $r = 3$  et  $n = 7$

Soit  $X_0 = (0.5, 0.5, 0.5)$ . Les figures 1 et 2 représentent une coupe de la fonction dans les plans  $(X_1, X_2)$  et  $(X_2, X_3)$ . Grâce à ces deux plans, nous pouvons localiser le minimum à l'aide des équations paramétriques des différents plans. Les figures 3 et 4 montrent le mauvais conditionnement du problème au minimum global  $(0.23, 0.55, 0.88)$  de la fonction.

# ITERATION 0

CONTOUR KEY	
1	-1.3
2	-1.0
3	-0.5
4	-0.9
5	0.5
6	1.0
7	1.7
8	2.5
9	3.5
10	4.5
11	5.5
12	6.5
13	7.2
14	7.8
15	8.5

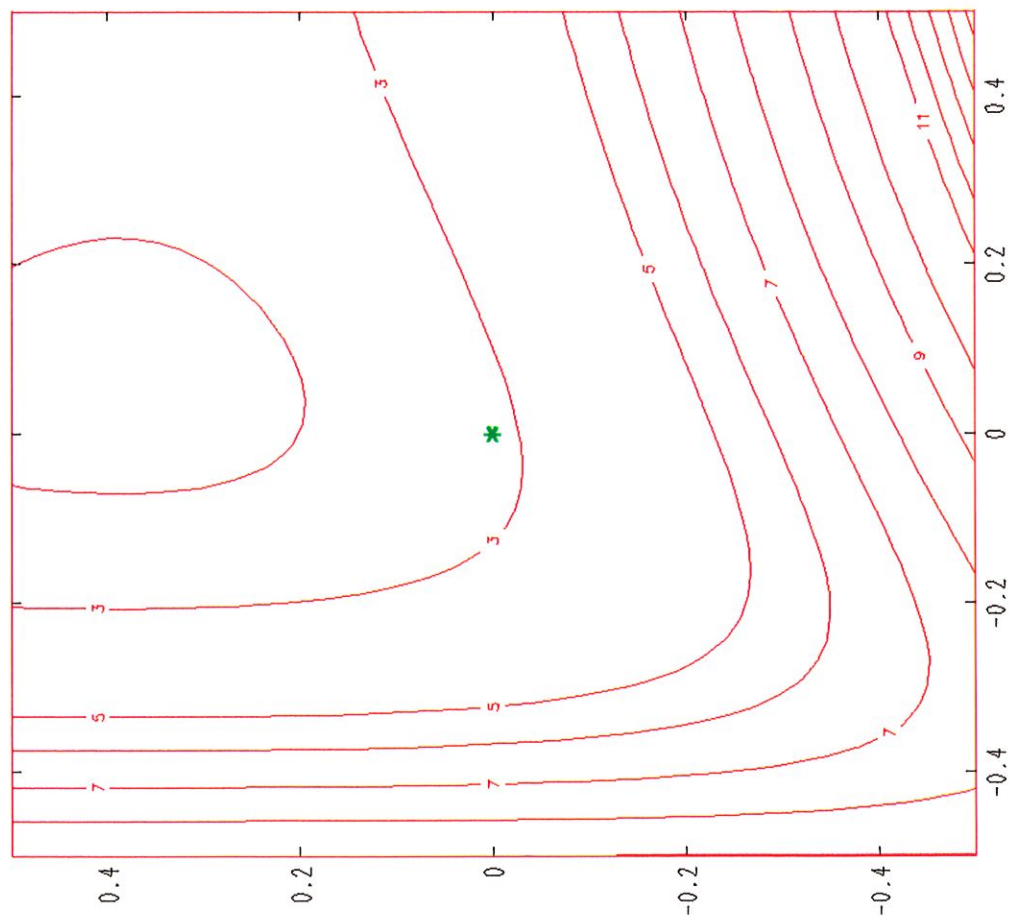
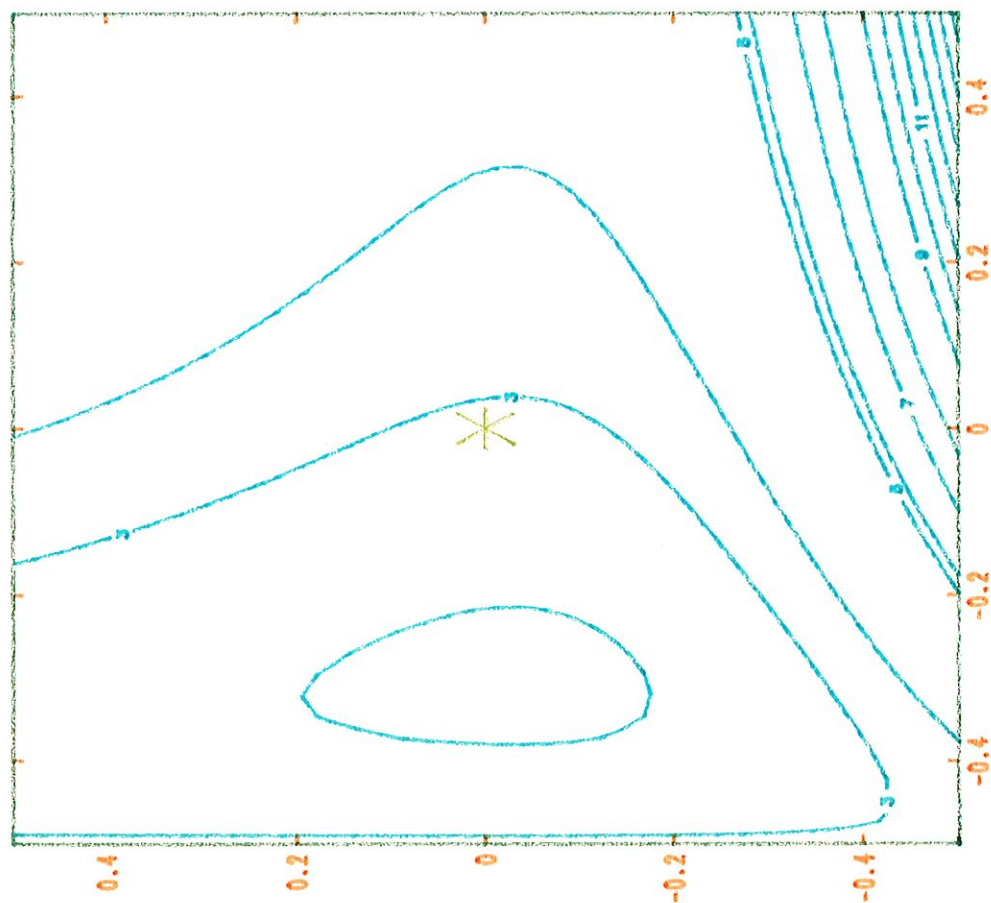


FIGURE 1

ITERATION

0

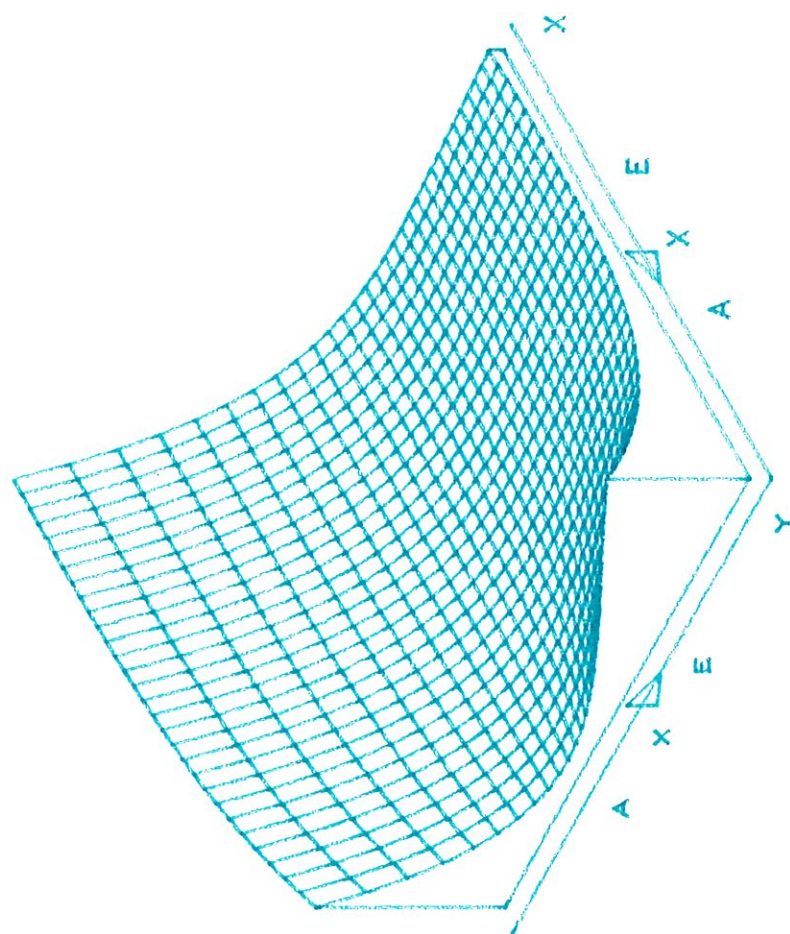


CONTOUR KEY	
1	-1.4
2	-1.0
3	-0.5
4	0.0
5	0.5
6	1.0
7	1.5
8	2.0
9	2.5
10	3.0
11	3.5
12	4.0
13	4.5
14	5.0
15	5.5

FIGURE 2

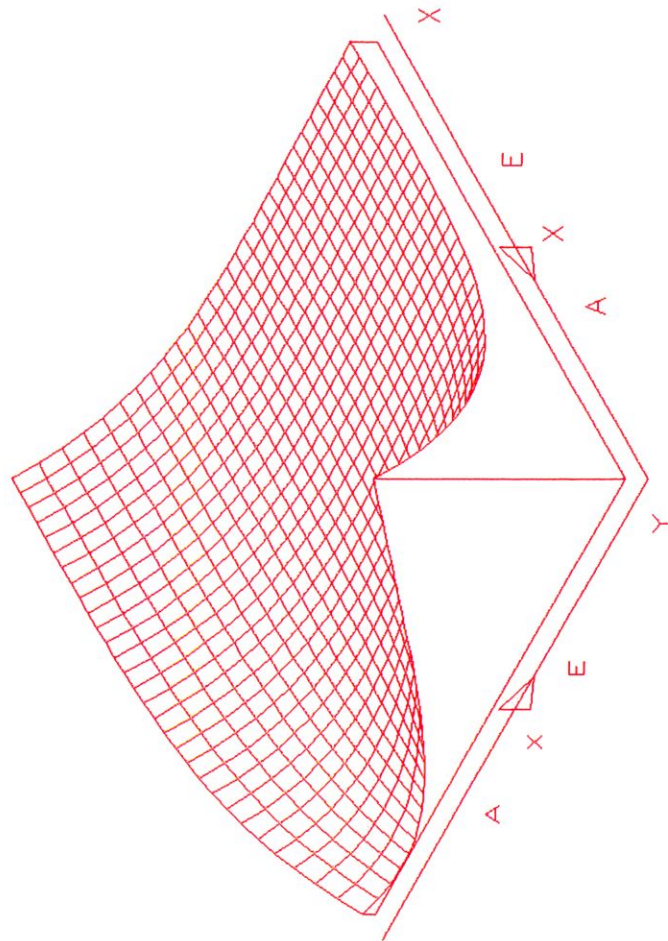
PLAN DE LA + GRANDE ET LA + PETITE VAL. PROPRE

FIGURE 3



# PLAN DES DEUX PLUS GRANDES VALEURS PROPRES

FIGURE 4





### 3.3. Conclusions.

Les tests effectués montrent qu'OPTIGRAPH répond aux trois formes d'aide énoncées au chapitre 1.

En ce qui concerne, l'aide au démarrage de la procédure de minimisation, c'est-à-dire au choix des valeurs initiales des paramètres, nous conseillons à l'utilisateur, de réaliser une ou plusieurs coupes dans les plans des variables qu'il considère comme prépondérantes. Il pourra ainsi localiser la position d'un ou plusieurs minimums.

Lors du suivi du point courant, la visualisation dans le plan des directions de descente, lui permettra le cas échéant de corriger le tir et d'accroître de fait la vitesse de convergence de l'algorithme ou d'éviter un minimum local de la fonction. En combinant cette visualisation à celle dans le plan des vecteurs propres, il peut aisément repérer les variables qui requièrent une modification et évaluer un nouveau point de départ pour la minimisation, à cet effet il utilisera les équations paramétriques des différents plans. Une visualisation de l'approximation quadratique dans le plan des vecteurs propres, associés à la plus petite et la plus grande valeur propre, lui fournit une idée du conditionnement de l'approximation de la matrice hessienne.

Lors de l'arrêt des calculs, nous conseillons à l'utilisateur des visualisations dans les plans des axes afin de tester la convergence de l'algorithme.



## Chapitre 4

#### 4. Informatisation du problème.

##### 4.0. Introduction.

Ce chapitre est consacré aux aspects informatiques du problème. Un logiciel de visualisation des résultats, baptisé OPTIGRAPH, a été développé au centre de calcul du laboratoire de recherche de Solvay S.A. à Neder-over-Hembeek.

OPTIGRAPH est conçu dans l'optique de pouvoir être utilisé par des chercheurs dispersés sur des sites différents. Aussi, nous avons dû respecter les exigences suivantes pour le logiciel :

- la convivialité
  - la portabilité
  - la robustesse
  - la lisibilité des programmes
- et leur documentation.

Afin de satisfaire à ces différentes exigences, nous avons fait appel, au maximum, à des outils normalisés tant au niveau des outils d'aide à la programmation qu'au choix du langage et des outils graphiques.

La robustesse est assurée par des algorithmes mathématiques qui ont fait leurs preuves (interpolation cubique

dans le cas des fonctions d'une variable et B.F.G.S.  
dans le cas des fonctions de plusieurs variables) et par des  
implémentations fiables extraites de la bibliothèque scientifique NAG ( Numérical Algorithms group ).

Le paragraphe suivant reprend plus en détail  
les concepts et idées qui viennent d'être exposés.

#### 4.1. Outils et moyens disponibles.

##### 4.1.1. Les outils logiciels.

Les outils logiciels sur lesquels nous nous sommes appuyés sont :

ANSI FORTRAN 77 : En matière de langage, le centre de calcul de Solvay a adopté comme standard l'ANSI FORTRAN 77. On rejette en outre les possibilités offertes par le VAX FORTRAN du constructeur ( Digital Equipement ) afin d'assurer aux logiciels une meilleure, voire totale, portabilité avec d'autres systèmes.

AGAPE : Aide à la Gestion d'Applications en mode Plein Ecran  
AGAPE est un logiciel développé par le Centre de Calcul qui génère sous forme de source FORTRAN, la phase d'acquisition de données. A partir d'écrans "dessinés" au moyen d'un éditeur, on définit les champs fixes d'une part et les champs variables d'autre part. Les champs variables se subdivisent en "input" et "output". Ces deux notions se définissent par rapport à l'écran. La routine générée pourra alors être utilisée telle quelle par le programmeur en passant les variables d'input et d'output en argument.

N.A.G. : Numerical Algorithms Group scientific library.

N.A.G. est une librairie de modules à orientation scientifique développée par la société britannique NAG en collaboration avec des spécialistes universitaires en applications software et en analyse numérique, et d'instituts de recherches. Les routines NAG, écrites en ANSI FORTRAN 77 et mises à jour régulièrement, couvrent un grand nombre de domaines des mathématiques tels les équations différentielles, l'optimisation de fonctions, le calcul matriciel et l'algèbre linéaire, les fonctions spéciales et les fonctions statistiques, ... Elles constituent pour le programmeur un outil robuste, performant et entièrement portable qui répond dans la plupart des cas à leurs besoins.

G.K.S. : Graphical Kernel System.

G.K.S. est un logiciel graphique normalisé, développé par l'université de Darmstadt (RFA), concernant le graphisme bidimensionnel indépendamment du type de matériel du poste graphique utilisé (écran, tablette graphique, table traçante, ...) Il permet en outre de sauvegarder sur des fiches normalisées (METAFILE) les instructions nécessaires à la réalisation d'un graphique et interprétables par tout site disposant du logiciel G.K.S. Ces fichiers serviront entre autres à débiter plus tard les graphiques au moyen de logiciels

appropriés sur des matériels ( écrans graphiques ou tables traçantes ) localisés en des endroits différents.

Caractéristiques principales de G.K.S. :

- simplicité des concepts de base
- normalisation de la programmation en matière graphique.
- portabilité et indépendance vis-à-vis du matériel
- norme ISO ( International Standard Organisation )
- transportabilité et archivage des informations graphiques normalisées.

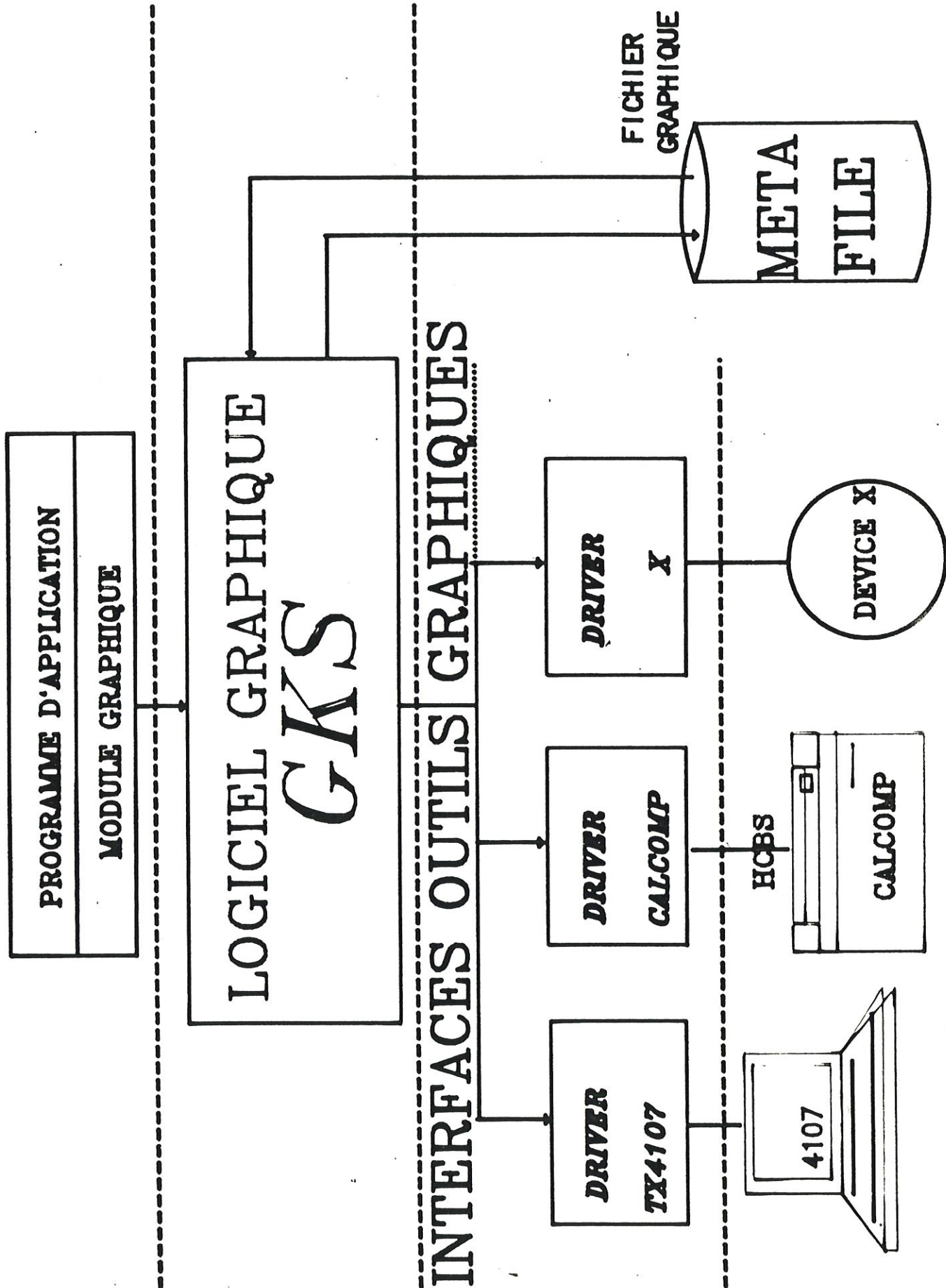
La figure suivante synthétise les caractéristiques principales de G.K.S.

Précisons le fait que G.K.S. ne fournit que des routines graphiques élémentaires. La réalisation des programmes de haut niveau permettant par exemple le tracé des vues isométriques ( ce qui dans la figure est repris sous le vocable MODULE GRAPHIQUE ) doit être envisagée comme un problème distinct. Nous avons sur ce point utilisé les possibilités graphiques offertes par la bibliothèque scientifique N.A.G. en les couplant à celles offertes par G.K.S.

#### 4.1.2. Le matériel.

Le centre de calcul de Solvay dispose de deux ordinateurs de la firme DIGITAL EQUIPMENT, Un VAX 11/780





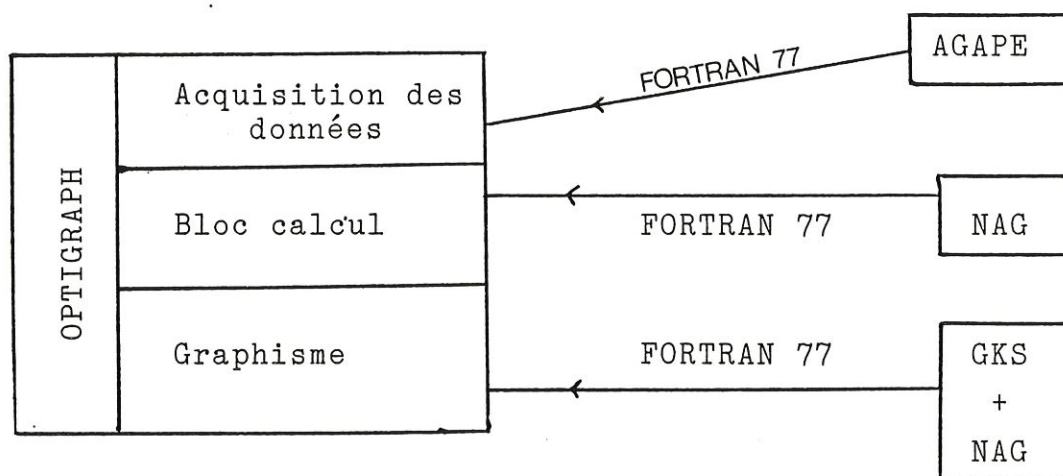
ET UN 8600. Ces deux ordinateurs sont couplés en ce sens qu'ils partagent les mêmes ressources "disque".

Les outils graphiques disponibles sont de type divers : écrans graphiques couleur noir et blanc ( Tektronix 4107, DIGITAL VT 240 et VT 241), imprimantes à jet d'encre (Tektronic ) et tables traçantes (Hewlett Packard, Calcomp) Tous sont utilisables par OPTIGRAPH.

#### 4.2. Méthodologie générale.

Nous avons synthétisé la méthodologie suivie pour réaliser le logiciel OPTIGRAPH sur la figure suivante.

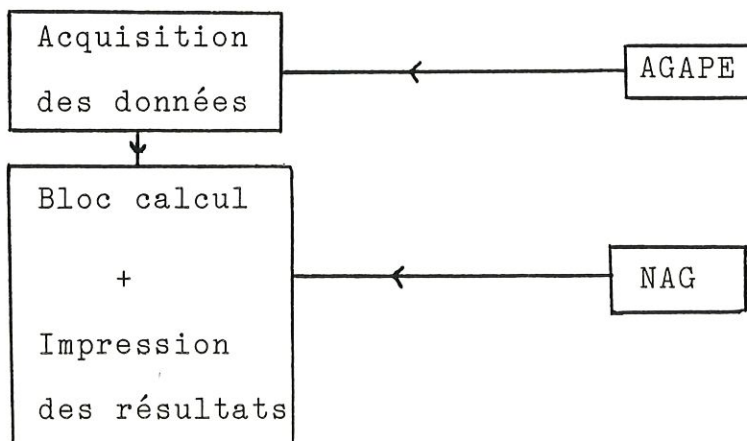
##### SCHEMA DE LA METHODOLOGIE UTILISEE POUR OPTIGRAPH.



OPTIGRAPH, quant à lui, peut s'intégrer dans tout programme d'optimisation au niveau de la routine d'impression des résultats.

Classiquement, un programme d'optimisation répond à l'organigramme suivant :

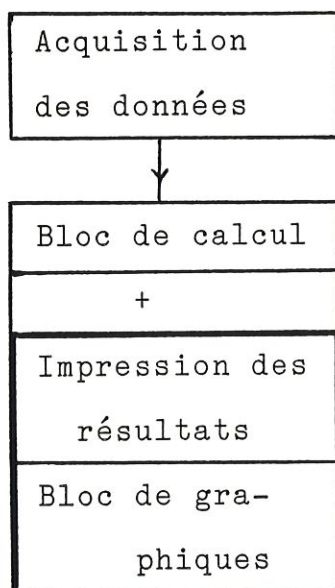
##### ORGANIGRAMME GENERAL D'UN PROGRAMME D'OPTIMISATION.



La routine d'impression des résultats est ainsi intégrée dans la routine implémentant l'algorithme de calcul.

Nous avons greffé notre logiciel graphique sur cette structure d'une manière naturelle ainsi que le montre la figure ci-dessous.

ORGANIGRAMME GENERAL D'UN PROGRAMME D'OPTIMISATION PRENANT EN COMPTE LE GRAPHISME.



Le nom donné au programme d'optimisation dans notre travail est QNVISU ( Quasi-Newton Visualisation ).

### 4.3. Le logiciel OPTIGRAPH.

#### 4.3.1. Contexte général.

Au vu de la portabilité et de la flexibilité qu'ont atteint ces dernières années les logiciels graphiques et numériques, nous avons pensé à créer un logiciel baptisé OPTIGRAPH où le graphisme peut aider l'utilisateur dans son interprétation des résultats de calcul.

De telles réalisations n'ayant jamais été effectuées, à notre avis, nous nous sommes efforcé de dégager une méthodologie générale.

OPTIGRAPH sera conçu dans une optique où la portabilité c'est-à-dire l'aptitude à fonctionner sur un grand nombre de systèmes est un facteur essentiel. Nos choix sont fixés sur les critères et outils suivants :

- la programmation en Fortan 77 standard
- les outils graphiques normalisés G.K.S
- la librairie scientifique NAG
- les outils d'aide à la saisie interactive des données et à la gestion d'application plein écran ( AGAPE ) mis au point chez Solvay.



A ce prix, nous serons assuré d'une très bonne robustesse et d'une très bonne portabilité tant au niveau ordinateur qu'au niveau matériels graphiques.

Les pages qui suivent sont consacrées à la description d'OPTIGRAPH. Nous tenons à rappeler que ce logiciel étant un prototype, il ne faut pas le considérer comme une fin en soi mais comme une première étape qui doit être consolidée par des tests intensifs.

#### 4.3.2. Présentation générale d'OPTIGRAPH.

OPTIGRAPH permet donc de représenter des informations numériques sous forme graphique.

Les graphiques proposés sont de deux types :

- courbes de niveaux
- vues isométriques.

Ils peuvent être obtenus ( étant donné le choix de la norme G.K.S. ) sur tout matériel graphique ( tables traçantes, écrans, ... ).

Les données nécessaires aux graphiques sont communiquées :

- soit par l'algorithme d'optimisation lui-même
- soit par calculs ou évaluations numériques complémentaires
- soit directement par l'utilisateur.



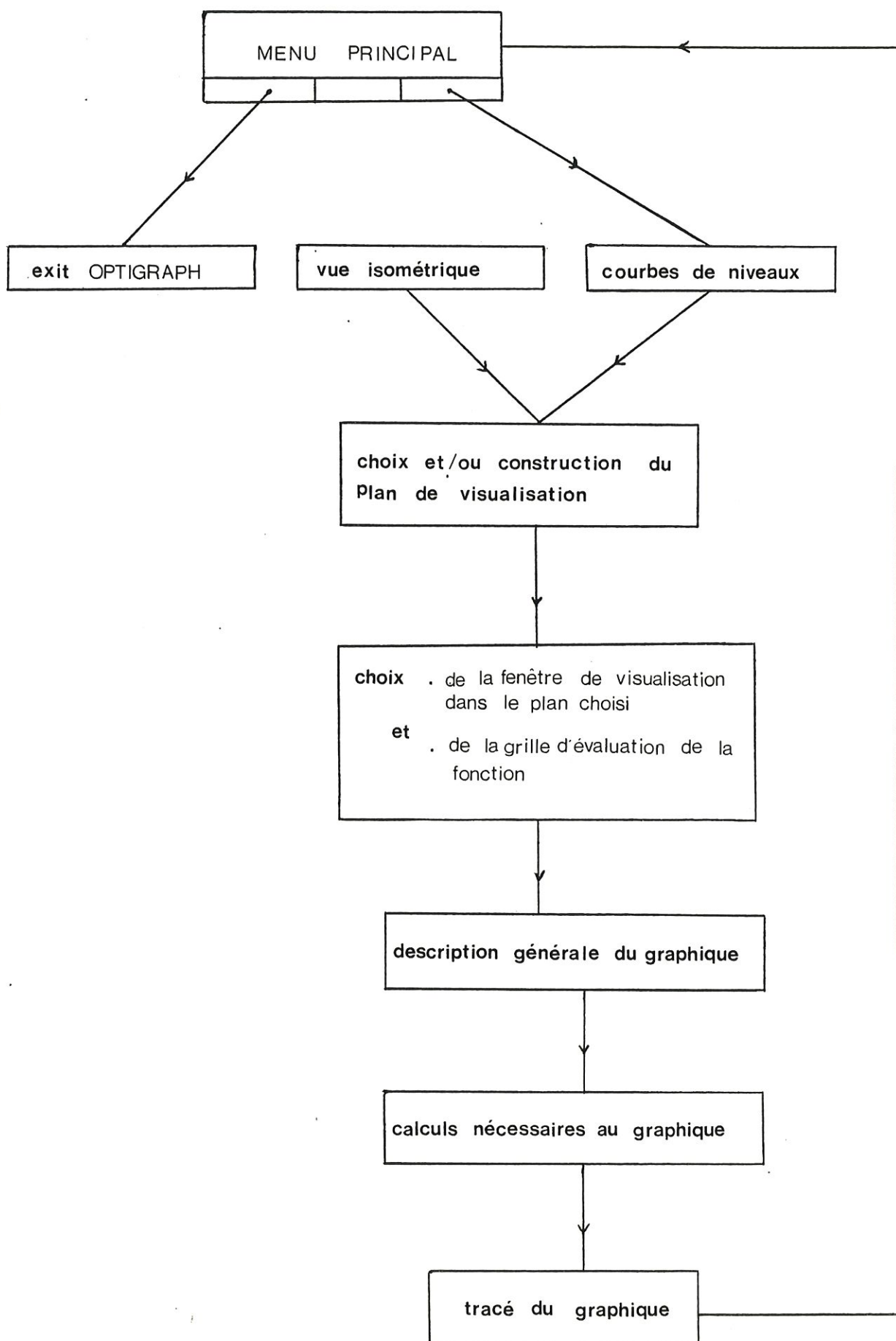
Les options du graphisme sont présentées sous forme de menus structurés en arbre; elles permettent une grande flexibilité.

Les informations nécessaires sont :

- une description générale ( titres et commentaires, libellés des axes, nombres des courbes, angle de vue pour la visualisation en trois dimensions
- le domaine de représentation (choix du plan de visualisation dans le cas d'une fonction de plusieurs variables, choix des échelles,...)
- une description fine de la fenêtre de visualisation ( quadrillage du pavé de visualisation )

La structure générale d'OPTIGRAPH est donnée par l'organigramme suivant :

## structure d'OPTIGRAPH



D'une manière générale, un ensemble cohérent de données constitue une grille écran; avant de passer à la description des différents écrans, voyons comment les utiliser.

Chaque écran est constitué de zones. On peut se déplacer dans une zone au moyen des flèches ← et → d'une zone à une autre par (PF1) suivi d'une des flèches.

Les caractères tapés dans une zone peuvent remplacer les caractères déjà existants ou s'insérer au milieu d'eux ( mode insertion).

La combinaison des touches (CTRL) (A) permet de passer d'un mode à l'autre.

Pour passer d'une zone à une autre et/ou pour quitter un écran les touches suivantes seront utilisées :

RETURN : après chaque entrée dans une zone de saisie.

RETURN ou PF3 : si on se trouve à la dernière zone de l'écran.

PF3 : dès que l'on veut sortir d'un écran sans modifier les valeurs affichées.

Toute exception à ces règles sera signalée lors de la description des écrans.

#### 4.3.3. Arbres des écrans.

Au début de son exécution , OPTIGRAPH présente un premier écran : le menu principal. Cette écran permet à l'utilisateur de faire choix entre cinq options :

##### Ecran 1 : Menu principal.

<p style="text-align: center;"><b><u>VISUALISATION</u></b></p> <p>1 : COURBES DE NIVEAUX DE LA FONCTION 2 : COURBES DE NIVEAUX DE L'APPROXIMATION QUADRATIQUE 3 : REPRESENTATION EN DIMENSION 3 DE LA FONCTION 4 : REPRESENTATION EN DIMENSION 3 DE L'APPROXIMATION QUADRATIQUE  0 : SORTIE DE LA VISUALISATION</p> <p style="text-align: right;">VOTRE CHOIX : <input checked="" type="checkbox"/> 0</p>
---

Après avoir fait son choix, on presse la touche RETURN. Si l'option 0 est choisie nous sortons directement d'OPTIGRAPH sinon suivant l'option désirée, des branches différentes de l'arbre des écrans sont parcourues ( Figure 1 : Arbre des écrans ).

Nous allons d'abord décrire les écrans apparaissant à la suite de l'option 1 ou 2 ( les données nécessaires au graphique fournies par l'utilisateur sont les mêmes pour ces deux options, qui ne se différencient qu'au niveau des opérations numériques )

Le deuxième écran n'apparaît en réalité que si la dimension de l'espace considéré est supérieure à deux. Il permet à l'utilisateur de désigner le plan de visualisation de la fonction c'est-à-dire le plan sur lequel la fonction va être évaluée pour le graphisme.

Ecran 2 : Choix du plan.

**CHOIX DU PLAN**

0 : PLAN DES DIRECTIONS DE DESCENTES  
 1 : PLAN DES DEUX PLUS GRANDES VALEURS PROPRES  
 2 : PLAN DE LA PLUS GRANDE ET DE LA PLUS PETITE  
     VALEUR PROPRE  
 3 : SI VOUS DESIREZ INTRODUIRE UN PLAN QUELCONQUE  
 4 : MEME PLAN QUE POUR LA VISUALISATION PRECEDENTE  
 5 : RETOUR AU MENU PRINCIPAL

VOTRE CHOIX = 0

Dans le cas d'une méthode Quasi-Newton, l'option 0 du deuxième écran correspond, en fait, au plan engendré par le gradient et la direction Quasi-Newton et passant par l'estimation du minimum communiqué par l'algorithme d'optimisation.

L'estimation du minimum, dans le cas des options 1 et 2 appartient au plan choisi.

Si l'option 4 de l'écran 2 est choisie, nous demandons à l'utilisateur de nous communiquer les équations paramétriques du plan qu'il désire. L'écran 2.1. se charge de cette tâche (cfr Figure 1 : Arbre des écrans ).



Ecran 2.1. : Définition du plan.

EQUATION PARAMETRIQUE DU PLAN			
COMPOSANTE	ORIGINE	DIRECTION 1	DIRECTION 2
1	0.2	1	0
2	0.9	.000000000	1.000000000
3	-5	.000000000	0

L'écran est constitué de plusieurs zones.

On peut se déplacer dans une zone à l'aide des flèches → et ←, d'une zone à la suivante au moyen de la touche (RETURN) et d'une zone à la précédente à l'aide de la touche (PF1) suivie de la flèche ← . Si on est en fin de ligne (PF4) permet de retourner à la première zone de la ligne précédente.

Si à l'écran 2, l'option 5 est choisie nous retournons à l'écran 1.

L'écran 3 auquel on accède via les options 0, 1, 2, 3 et 4 de l'écran 2 permet à l'utilisateur de faire le choix du pavé du plan sur lequel la fonction sera visualisée et de la grille ( quadrillage du pavé ) d'évaluation de la fonction . Ou bien il peut reprendre une grille existante, ou bien il peut définir une nouvelle grille.



Ecran 3 : Choix du domaine de visualisation  
et de la grille.

**CHOIX DE LA GRILLE**

0 : NOUVELLE GRILLE  
 1 : MEME GRILLE QUE POUR LA VISUALISATION PRECEDENTE  
 2 : LA GRILLE ET LES VALEURS DE LA FONCTION ASSOCIEES  
 A LA GRILLE DANS LE PLAN CONSIDERE SE TROUVENT  
 DANS UN FICHIER  
 3 : RETOUR AU MENU PRINCIPAL

**VOTRE CHOIX ?**    0

L'option 0 de l'écran 3 permet à l'utilisateur de communiquer son choix du pavé de représentation de la fonction dans le plan désigné à l'écran 2 ainsi que de définir la grille d'évaluation de la fonction à adopter sur le domaine de visualisation.

A une itération donnée, lorsque l'utilisateur a déjà effectué une visualisation, s'il désire conserver le même domaine et la même grille que pour la visualisation précédente ( celle qu'il vient d'effectuer ), alors l'option 1 de l'écran 3 lui évite de réintroduire ces données.

L'option 2 de l'écran 3 permet de reprendre une description antérieure d'un graphique dans un fichier. L'utilisateur devra communiquer son nom.

L'option 3 de l'écran 3 engendre un retour à l'écran 1.

L'écran 4.1. apparaît si l'utilisateur a choisi l'option 0,1 ou 2 à l'écran 3 et permet le sauvetage sur fichier de certaines données de la visualisation précédente.

L'écran 4.2. n'est accessible que si l'utilisateur choisit l'option 0 à l'écran 3.

Ecran 4.1. : Sauvetage des données.

DESIREZ-VOUS SAUVER LA GRILLE ET LES VALEURS DE LA  
FONCTION EVALUEE SUR LA GRILLE DE LA VISUALISATION  
PRECEDENTE (0=NON,1=OUI) ?

SAVE = 0

Si l'option 1 est choisie à cet écran, l'utilisateur est invité à fournir le nom du fichier.

Ecran 4.2. Définition du Domaine de représentation et de la grille.

**DEFINITION DE LA GRILLE**

DONNEZ LA BORNE INFERIEURE ET SUPERIEURE DE LA GRILLE  
PAR RAPPORT A L'AXE X

XMIN = -7  
XMAX = 10

DONNEZ LA BORNE INFERIEURE ET SUPERIEURE DE LA GRILLE  
PAR RAPPORT A L'AXE Y

YMIN = -5  
YMAX = 10

DONNEZ LE NOMBRE DE DROITES VERTICALES ET HORIZONTALES  
FORMANT LE QUADRILLAGE DE LA GRILLE

NV = 10  
NH = 15

Ecran 5 : Choix du nombre et de la hauteur des  
courbes de niveaux.

**CHOIX DES COURBES DE NIVEAUX**

- SPECIFIER LE NOMBRE DE COURBES DE NIVEAUX

NBRE = 15

- CHOIX DE LA VALEUR DES COURBES DE NIVEAUX

- 0 : SI LES COURBES DOIVENT ETRE REPARTIES DU  
MINIMUM AU MAXIMUM DE LA FONCTION SUR LA GRILLE
- 1 : SI VOUS DESIREZ FOURNIR LA HAUTEUR DES COURBES  
DE NIVEAUX
- 2 : SI VOUS DESIREZ FOURNIR LA HAUTEUR MAXIMALE ET  
MINIMALE DES COURBES DE NIVEAUX

VOTRE CHOIX = 1

Si l'option 0 est choisie à l'écran 5 alors l'étendue entre la valeur minimale et maximale de la fonction sur la grille est divisée en N parties égales où N est le nombre de courbes de niveaux, et les hauteurs des courbes de niveaux sont définies comme le point milieu des différents intervalles.

L'option 1 de l'écran 5 permet à l'utilisateur de communiquer la hauteur qu'il désire pour chaque courbe de niveaux. L'opération est réalisée à l'aide de l'écran 6.

Quand l'option 2 est choisie à l'écran 5, l'écran 7 apparaît pour permettre à l'utilisateur de fournir la hauteur maximale et minimale qu'il désire pour les courbes de niveaux. Si N est le nombre de courbes de niveaux désiré, les (N-2) autres courbes seront définies à espaces réguliers dans l'étendue de la hauteur de la fonction.

Ecran 6. Hauteur des courbes de niveaux.

<u>CHOIX DE LA HAUTEUR DES COURBES DE NIVEAU</u>	
<u>NUMERO DE LA COURBE</u>	<u>VALEUR DE LA COURBE</u>
1	-2.0000000000
2	-1.5000000000
3	-1.0000000000
4	.0000000000
5	2.0000000000
6	4.0000000000
7	5.0000000000
8	6.0000000000
9	7.0000000000
10	15.0000000000

L'écran 6 est constitué de plusieurs zones.  
On se déplace d'une zone à la suivante à l'aide de la touche (RETURN) et d'une zone à la précédente à l'aide de la touche (PF 4).

Ecran 7 : Hauteurs maximales et minimales des  
courbes de niveaux.

DEFINITION DE LA VALEUR MAXIMALE ET MINIMALE DES COURBES DE NIVEAUX

HAUTEUR MINIMALE = -2.00000000

HAUTEUR MAXIMALE = 200

Si l'option 2 ou 3 a été choisie à l'écran 1 (menu principal ) alors les écrans 2,2.1,3,4.1,4.2 apparaissent dans le même ordre et sous les mêmes conditions. Après ces écrans, apparaîtra l'écran 8 qui permet à l'utilisateur de communiquer les angles de vue de la représentation isométrique d'une fonction.



Ecran 8 : Angles de vue.**DEFINITION DES ANGLES DE VUE DE LA REPRESENTATION  
EN DIMENSION TROIS****SPECIFIEZ L'ELEVATION DE L'AXE DE VUE PAR RAPPORT A LA BASE PLANE****THETA = 35.00****SPECIFIEZ L'ANGLE DE ROTATION AUTOUR DE L'AXE VERTICAL****PHI = 45.00**

Les valeurs THETA = 35 et PHI = 45 sont en général de bonnes valeurs.

Dans la section 4.3.4. qui suit, nous donnons la structure algorithmique générale d'OPTIGRAPH.

Le lecteur intéressé par plus de détails, peut consulter le code source du logiciel qui est largement autodocumenté.

( Annexe 2 ).





#### 4.3.4. Structure algorithmique du logiciel OPTIGRAPH.

L'exposé qui suit s'appuie sur la figure 2 donnant l'organigramme du logiciel.

Etape 1 : OPTIGRAPH commence par demander à l'utilisateur son choix de visualisation grâce à la routine SCCHVI qui génère l'écran 1, celui du menu principal. L'option désirée est retenue dans une variable baptisée choix.  
Si choix = 0, nous sortons d'OPTIGRAPH sinon nous passons à l'étape 2.

Etape 2 : Cette deuxième étape est consacrée à la définition du plan dans lequel la visualisation d'une fonction est demandée. Toute ce qui concerne le plan est réalisé par la routine CRPLAN. Ses rôles principaux sont :

- de permettre à l'utilisateur de communiquer le choix de son plan par l'intermédiaire de l'écran 2 ( choix du plan )
- de calculer ou de demander à l'utilisateur (écran 2.1. : Définition du plan), si nécessaire, les équations paramétriques du plan.

- D'orthonormer les directions du plan ( car les routines graphiques demandent un domaine rectangulaire ).

Les variables de sortie, principales de CRPLAN sont :

- A : point du plan.
- DIR1: une direction du plan.
- DIR2 : l'autre direction du plan, orthogonale à la première.
- ICHPL : le numéro associé au plan choisi par l'utilisateur.

Si ICHPL = 5 nous retournons à l'étape 1.

Etape 3 : Impression des équations paramétriques du plan choisies par l'utilisateur. L'opération est effectuée par la routine SAVEPL.

Etape 4 : Cette étape est consacrée à la définition du pavé du plan sur lequel la fonction sera visualisée et de la grille ( quadrillage du pavé ) définissant les différents points d'évaluation de la fonction. La routine GRILLE débute par la présentation de l'écran 3 : choix du domaine de visualisation et de la grille.  
Soit ICHGR, l'option choisie par l'utilisateur à cet écran.

Suivant la valeur d'ICHGR , GRILLE nous demandera de fournir :

si ICHGR = 0 - XMIN,XMAX,YMIN,YMAX : paramètres définissant le pavé.

- NV et NH respectivement le nombre de droites verticales et horizontales définissant le quadrillage du pavé.

- ILIFI = 1 : pour signaler que la lecture a été effectuée par écran.

si ICHGR = 1 - XMIN,XMAX,YMIN,YMAX,NV, NH gardent les mêmes valeurs que pour la visualisation précédente.

- ILIFI = 1 indique que la lecture n'a pas été effectuée sur fichier.

si ICHGR = 2 - XMIN,XMAX,YMIN,YMAX,NV,NH comme pour ICHGR = 0.

- HTS : tableau contenant les valeurs de la fonction, évaluée sur la grille
- ILIFI = 0, indique que les données ont été lues sur fichier.

si ICHGR = 3 Nous retournons à l'étape 1.

Etape 5 : Lectures et calculs des dernières données nécessaires au graphique et tracé.

. Si choix = 1

. Si ILIFI  $\neq$  0

Comme nous l'avons vu précédemment l'algorithme peut parfois travailler dans un espace réduit, soit N la dimension de cet espace et NINIT celle de l'espace des variables.

. Si  $N < NINIT$  alors l'utilisateur à l'étape 2 nous a fourni N équations pour définir le plan de visualisation, or le plan dans l'espace réel de minimisation requiert NINIT équations paramétriques. Une routine AGPLAN replace automatiquement le problème dans l'espace réel. L'adaptation du point et des directions orthonormales du plan se fait sur base des relations :

. Si la  $j^{i\text{ème}}$  variable  $X(j)$  est libre, c'est-à-dire appartient à l'espace réduit, alors

$$AA(j) = A(j)$$

$$DDIR\ 1(j) = DIR1(j)$$

$$DDIR\ 2(j) = DIR1(j)$$



Par contre, si la  $j^{\text{ième}}$  variable  $x(j)$  est fixée alors

$$AA(j) = X(j)$$

$$DDIR1(j) = 0$$

$$DDIR1(j) = 0$$

Dès que le plan a été évalué, nous calculons les valeurs de la fonction sur la grille et les stockons dans HTS grâce à la routine CALFUN.

Par contre, si  $N = NINIT$  nous appelons directement CALFUN pour évaluer les valeurs de la fonction sur la grille et les stocker dans HTS.

La routine NIVEAU est alors appelée. Elle appelle l'écran 5 par l'intermédiaire duquel on choisit le nombre et la hauteur des courbes de niveaux.

Puis, si nécessaire, on passe à l'écran 6 ( hauteur des courbes de niveaux ) ou à l'écran 7 ( hauteur maximale et minimale des courbes de niveaux et enfin, on effectue le tracé du graphique.

Les équations paramétriques du plan sont imprimées à l'écran pour permettre à l'utilisateur s'il le désire de calculer un nouveau point de départ (SAVEPL) c'est-à-dire d'intervenir directement dans l'algorithme de minimisation.

On en a alors terminé avec la visualisation et on retourne au menu principal c'est-à-dire à l'étape 1.



. Si choix = 2.

. Si ILIFI  $\neq$  0

La routine CALAQ évalue les valeurs de l'approximation quadratique de la fonction sur la grille et les stocke dans HTS.

. La routine NIVEAU exécute les différentes fonctions expliquées ci-dessus (cfr : CHOIX = 1 )

. On retourne au menu principal ( étape 1 ).

. Si choix = 3.

. Si ILIFI  $\neq$  0

. Si N < NINIT

La routine AGPLAN augmente les équations paramétriques du plan de l'espace réduit à l'espace initial.

CALFUN calcule et stocke dans HTS les valeurs de la fonction évaluée sur la grille

. Par contre, si N = NINIT, nous appelons directement la routine CALFUN pour évaluer sur la grille les valeurs de la fonction et les stocker dans HTS.

. La routine DIM3 demande les angles de vue de la représentation isométrique de la fonction à l'aide de l'écran 8 ( angles de vue ) et trace le graphique.

. La routine SAVEPL imprime à l'écran les équations paramétriques du plan pour permettre à l'utilisateur, s'il le désire d'évaluer un

nouveau point de départ pour relancer la minimisation.

. On en a alors terminé avec la visualisation et on retourne au menu principal.

. Si choix = 4

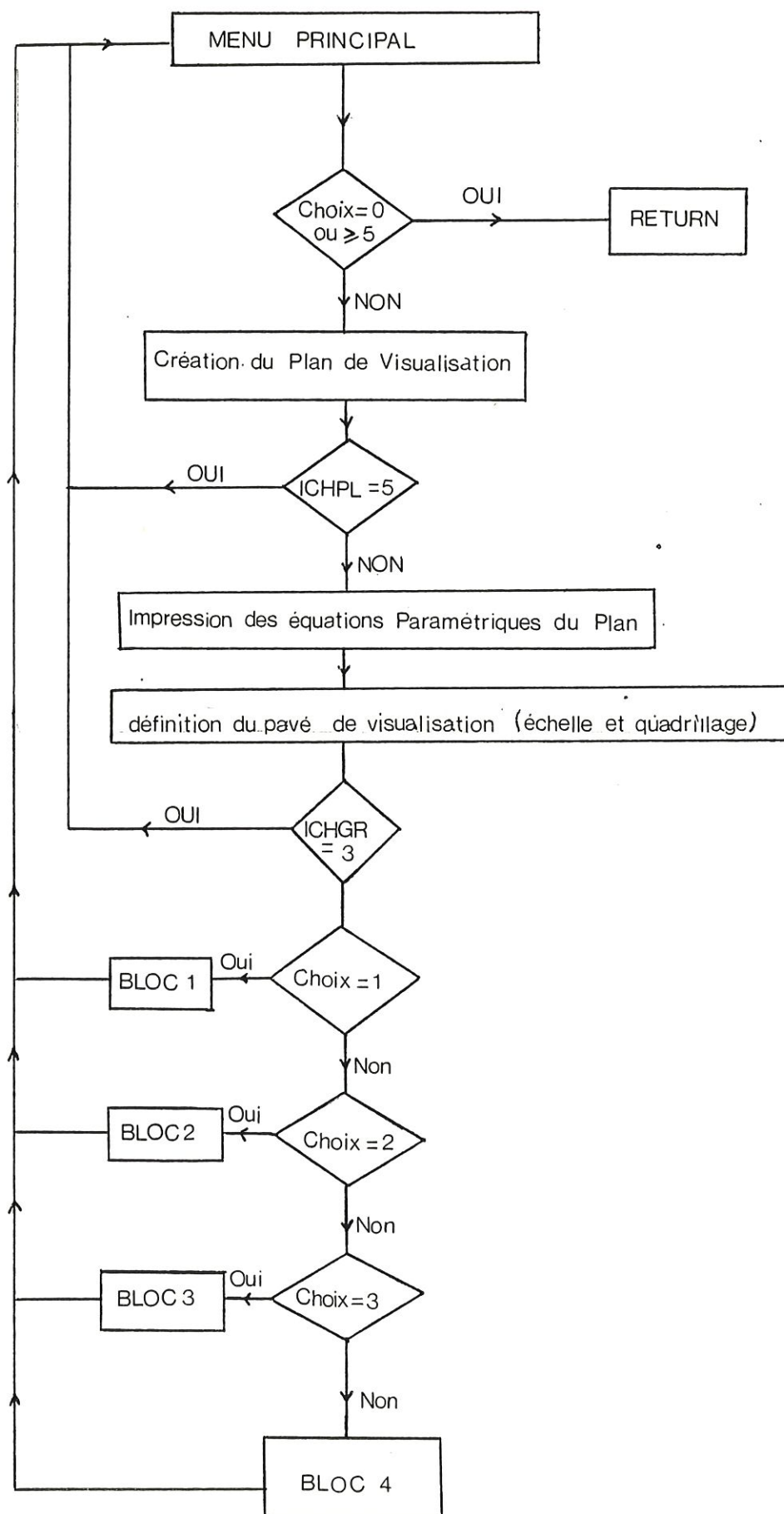
. Si ILIFI  $\neq$  0

. La routine CALAQ évalue les valeurs de l'approximation quadratique de la fonction sur la grille et les stocke dans HTS.

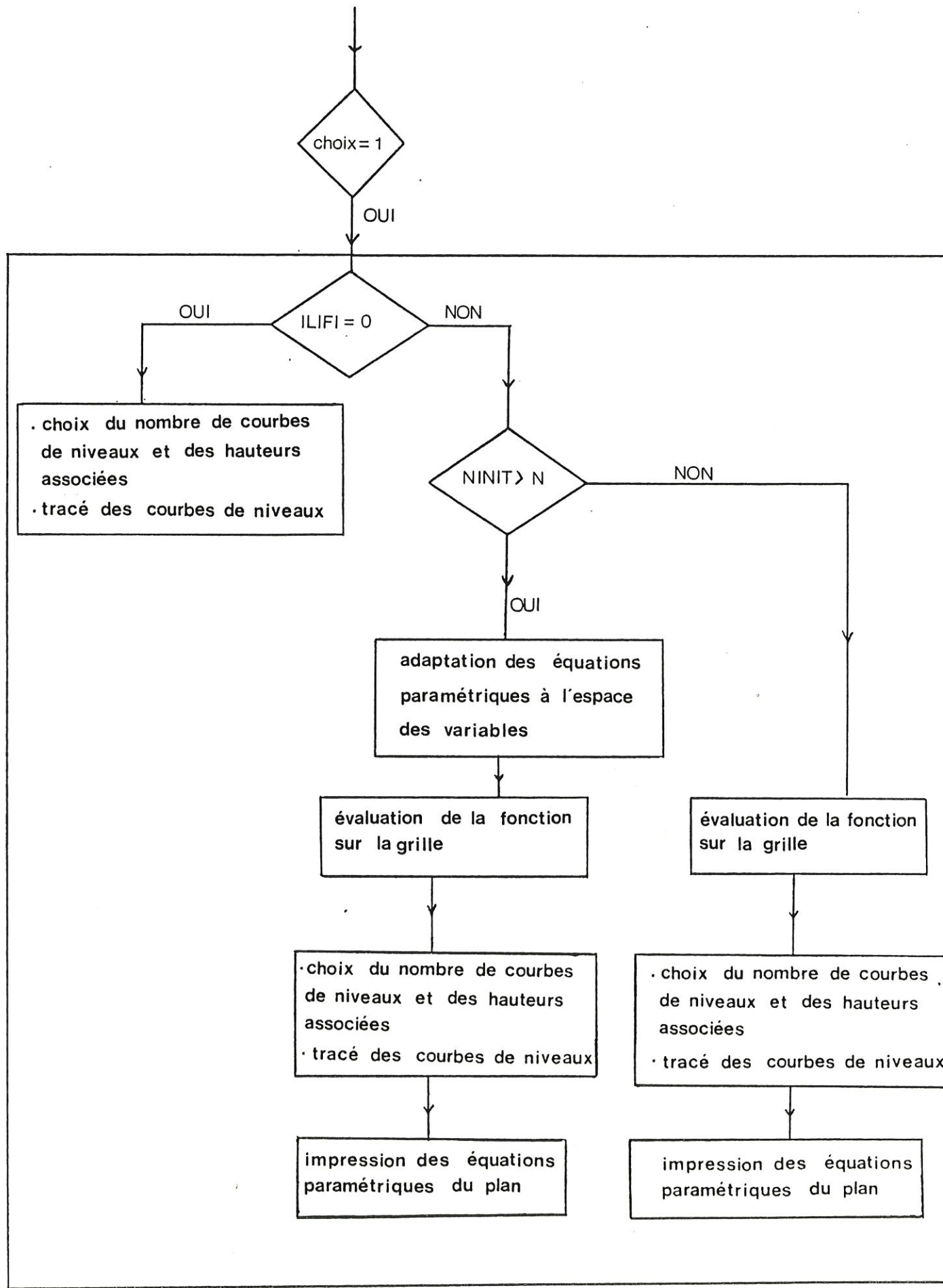
. La routine DIM3 exécute les différentes fonctions expliquées ci-dessus ( cfr CHOIX = 3 ).

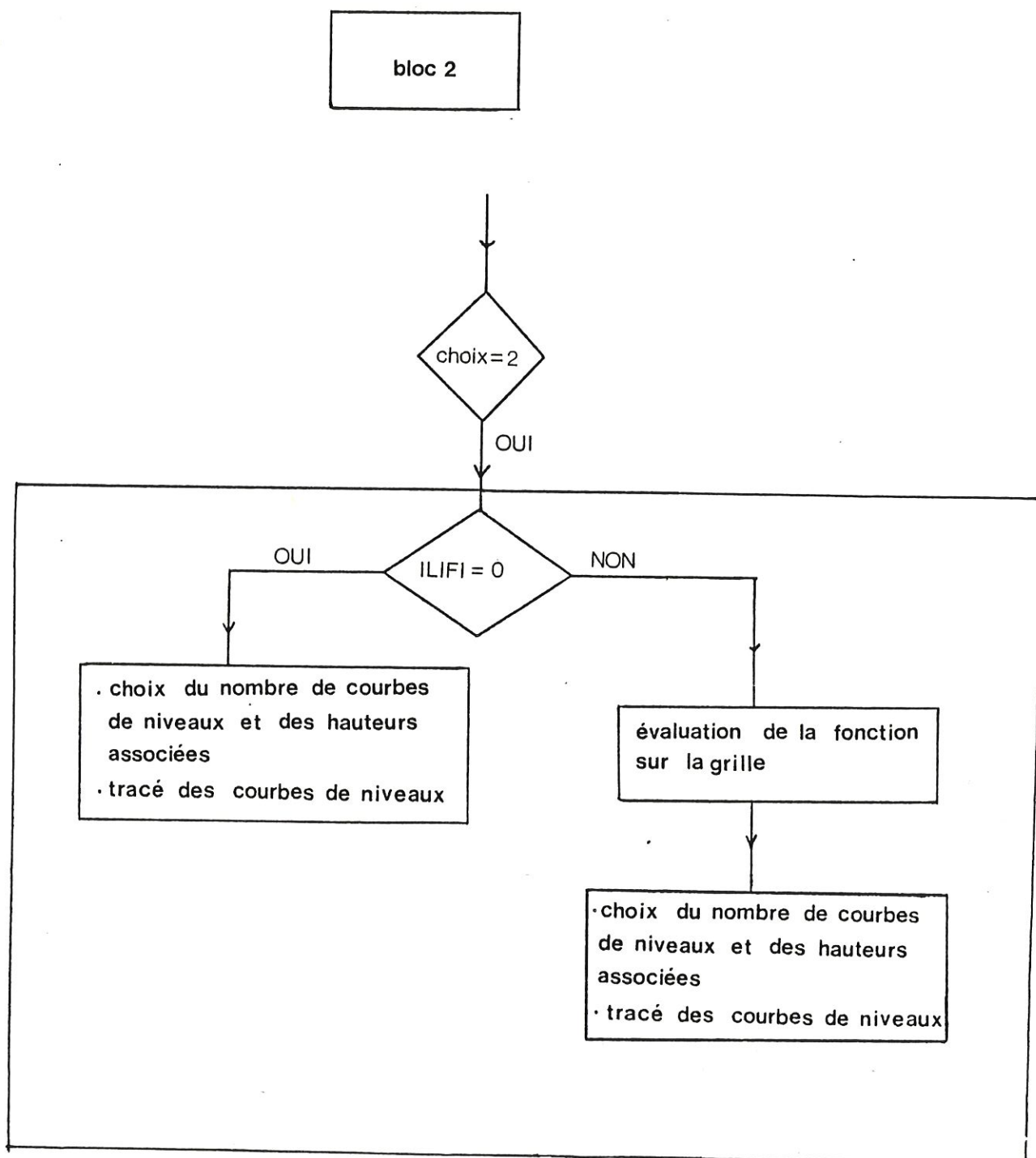
. On retourne au menu principal.

## — organigramme d'OPTIGRAPH —

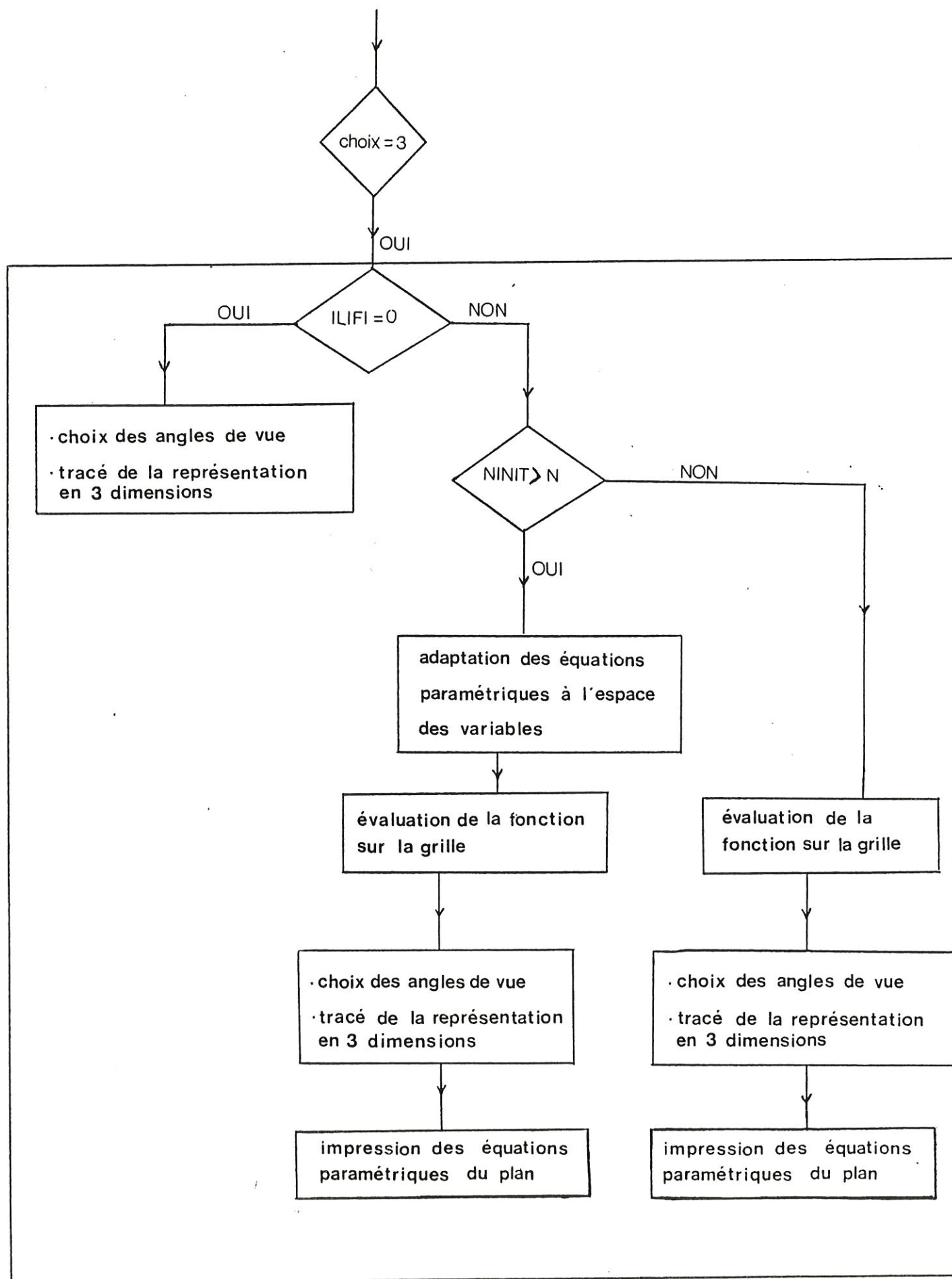


bloc 1

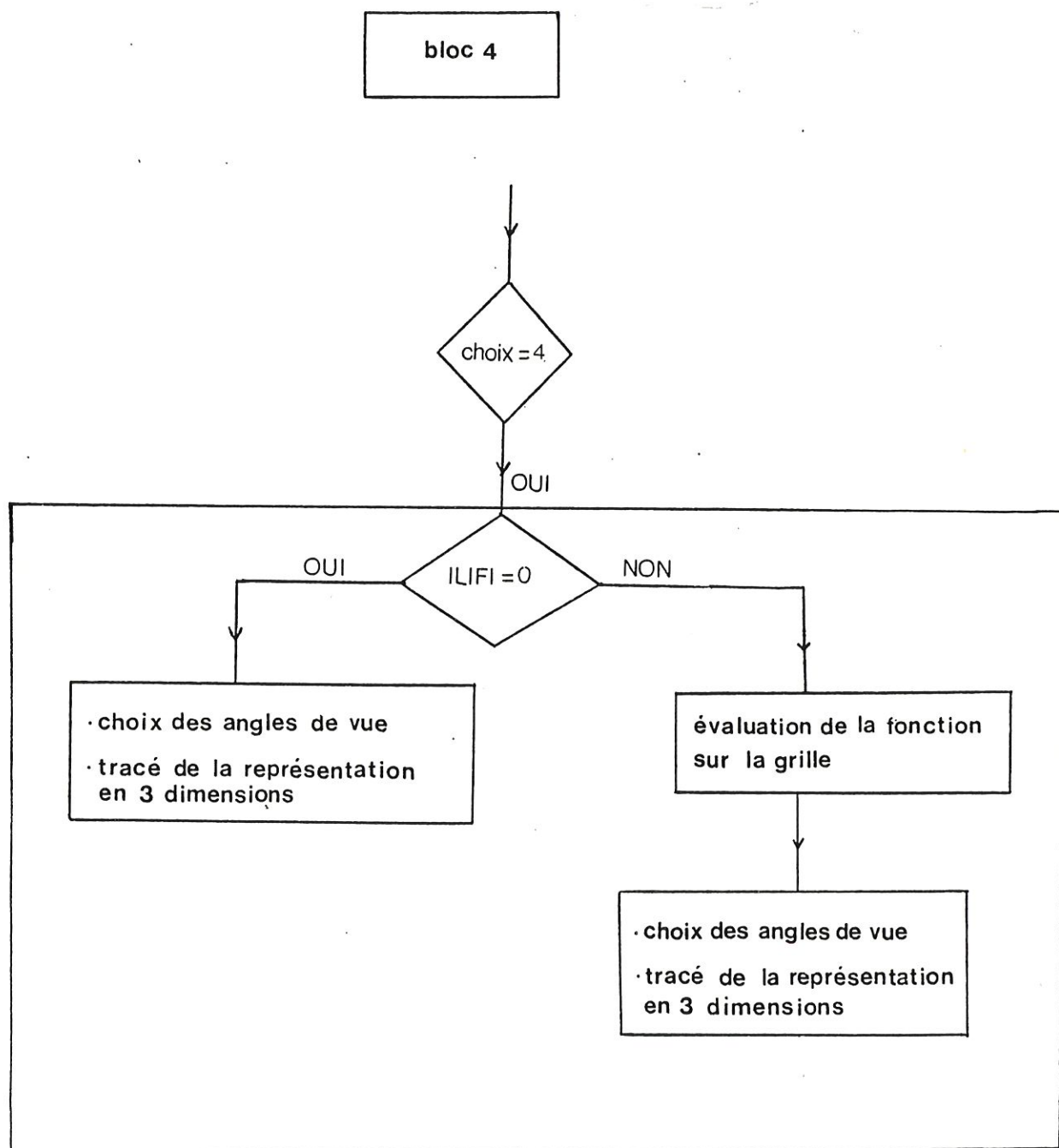




bloc 3







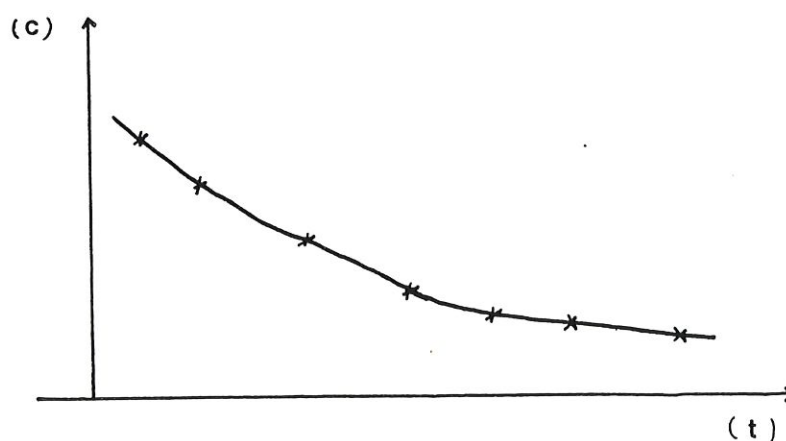
## Chapitre 5

## 5. Application à un problème de cinétique chimique.

### 5.0. Position du problème.

Il s'agit d'un problème couramment rencontré au laboratoire de recherches de Solvay à savoir :

- Ajuster un modèle théorique à un ensemble de données expérimentales. Ces dernières sont des relevés de concentrations (c) d'une substance chimique en fonction du temps (t). Voir figure ci-dessous.



Le modèle théorique proposé est du type exponentiel

$$g(t_1, k_1, k_2, c_1, c_2) = k_1 e^{-c_1 t} + k_2 e^{-c_2 t}$$

Il s'agit d'identifier les paramètres  $k_1, k_2, c_1$  et  $c_2$ .

La fonction à optimiser choisie est du type moindres carrés.

Le problème peut être formulé à présent :

Rechercher un quadruplet  $(k_1^*, k_2^*, c_1^*, c_2^*)$

minimisant l'expression

$$\sum_{i=1}^m (y_i - g(t, k_1, k_2, c_1, c_2))^2$$

où  $m$  désigne le nombre de couples de points expéri-

mentaux  $(t_i, y_i)$ .

### 5.1. Résultats du calcul.

Le minimum physique de la fonction est atteint en  $(k_1^* = 0.029, k_2^* = 0.050, c_1^* = 0.091, c_2^* = 0.001)$

La figure 1 montre que ce minimum est franc et que la fonction est "plate" ( cfr valeurs dans le "contour key" ).

Les figures 2 et 3 nous donnent des renseignements plus précis quant au conditionnement de la fonction au voisinage du minimum, les échelles des axes des coordonnées sont mille fois plus grandes que dans la figure 1.

Dans la figure 2 le plan des axes des coordonnées est celui des vecteurs propres correspondant aux plus grandes valeurs propres.

Dans la figure 3 c'est celui des vecteurs propres correspondant aux valeurs propres extrêmes.

La figure 4 donne une visualisation du même type que la figure 2 mais avec des échelles cent fois plus grandes ( c'est à dire  $10^5$  fois plus grandes que celles de la figure 1 ).

De l'analyse de ces graphiques, on peut conclure que plusieurs types de représentations sont nécessaires pour parvenir à des conclusions intéressantes à savoir :

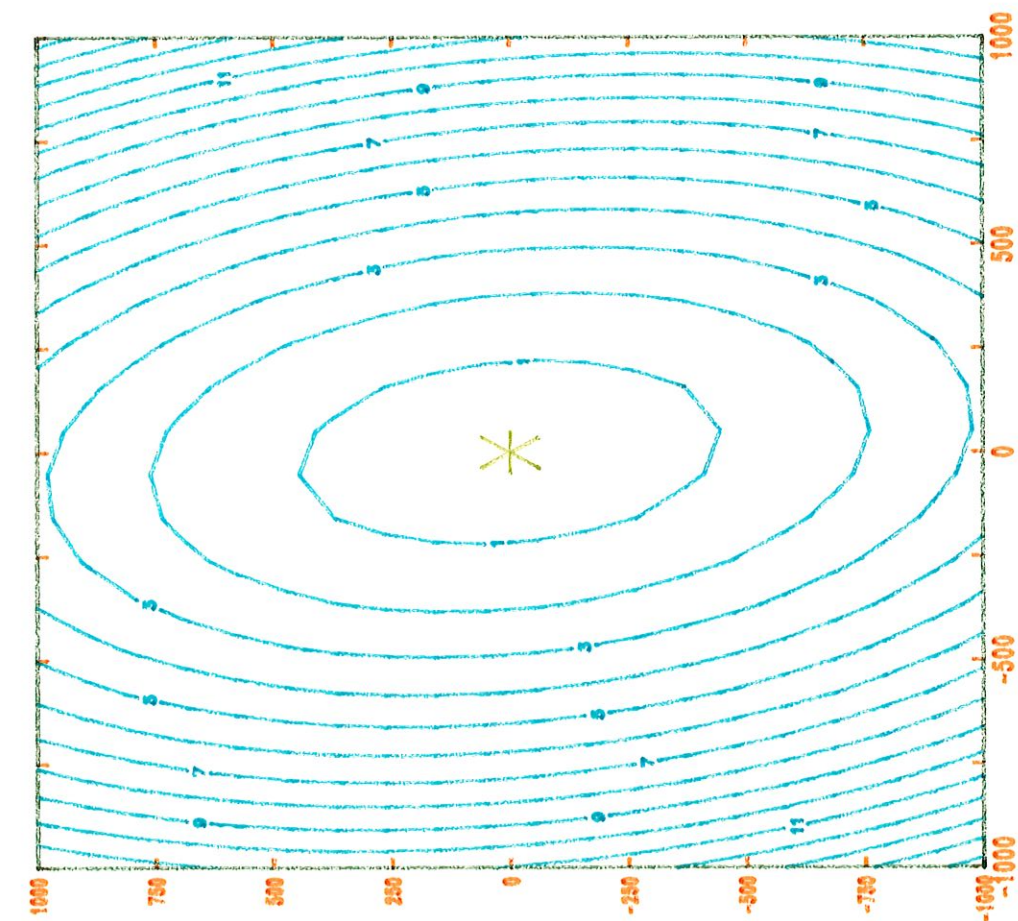
- 1 -que le minimum est franc.
- 2 -que la fonction à l'extrémum est "plate".
- 3 -que le problème est fort mal conditionné  
et dans quelle mesure il l'est.

En particulier, on remarquera que le mauvais conditionnement du problème rend pratiquement illusoire la possibilité de trouver un point de départ "optimal".



# ITERATION 16

PLAN DES DIRECTIONS DE DESCENTE



CONTOUR KEY	
1	0.0
2	0.0
3	0.0
4	0.0
5	0.0
6	0.0
7	0.0
8	0.0
9	0.0
10	0.0
11	0.0
12	0.0
13	0.0
14	0.0
15	0.0

FIGURE 1

# PLAN DES DEUX PLUS GRANDES VALEURS PROPRES

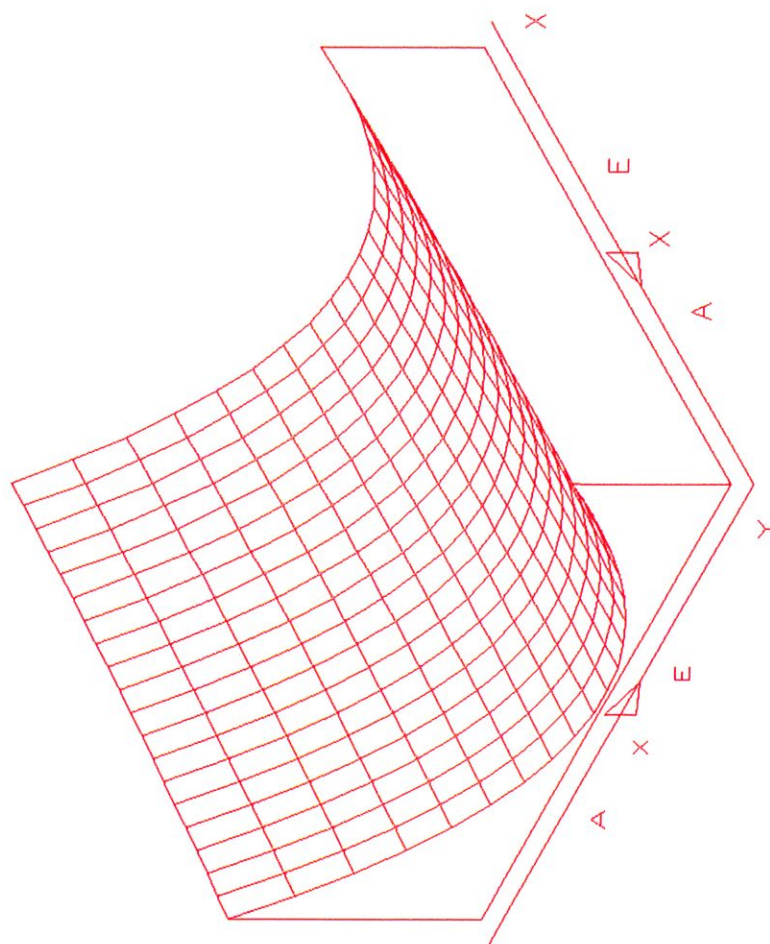
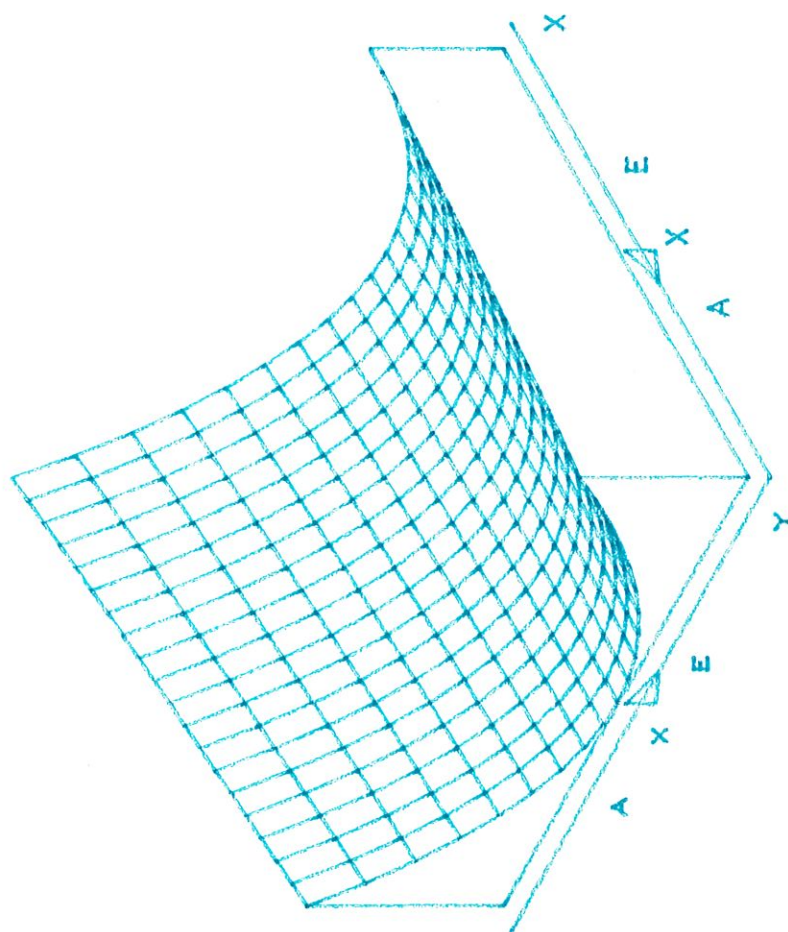


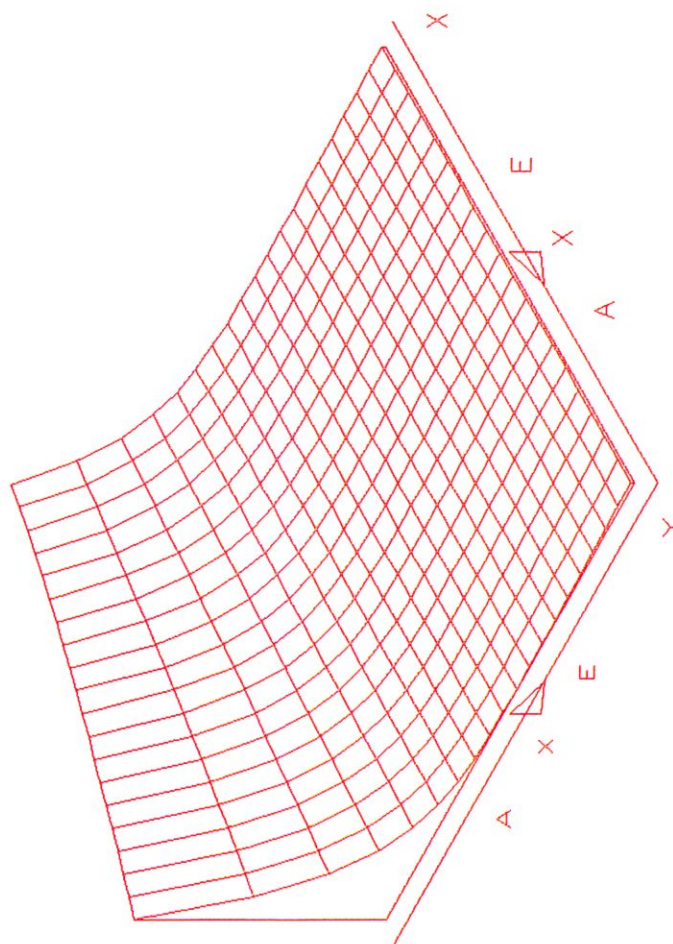
FIGURE 2

FIGURE 3



# PLAN DES DEUX PLUS GRANDES VALEURS PROPRES

FIGURE 4



## **Conclusions**



### CONCLUSIONS GENERALES.

En partant de l'idée d'introduire des techniques graphiques dans les algorithmes d'optimisation de fonctions réelles de variables réelles, nous nous étions fixés trois objectifs.

- fournir une aide au démarrage de la procédure de calcul;
- suivre la progression du point courant pendant l'exécution de l'algorithme afin de pouvoir, le cas échéant, en modifier la stratégie;
- compléter l'information numérique fournie lors de l'arrêt des calculs.

Le premier de ces trois points a déjà permis par des visualisations appropriées de bien maîtriser l'allure de la fonction au point de départ de l'optimisation, en particulier de mettre d'emblée en évidence des extrémums locaux. On a pu éviter ainsi nombre d'essais numériques dont la seule utilité était de s'assurer du minimum global .

Le suivi du point courant, si l'on veut en tirer un parti maximum, nécessite la maîtrise de l'algorithme. Ainsi, les algorithmes quasi-Newton que nous avons traités dans ce travail nous fournissent en plus de la position de l'itéré (point courant) :



- le gradient de la fonction en ce point,
- la direction de descente "quasi-Newton".
- une approximation du hessien grâce à laquelle nous déduisons certains renseignements sur la convexité de la fonction.

Toutes ces informations ont été exploitées au niveau des visualisations ( plan de descente "gradient-direction quasi-Newton", plan des vecteurs propres du hessien correspondant aux plans grandes valeurs propres ,...)

Les exemples traités dans ce travail montrent tout le parti que l'on peut tirer du graphisme; en particulier, on peut, si besoin en est, décider de modifier la stratégie de l'algorithme en introduisant des éléments d'appréciation déduits des visualisations.

Le dernier objectif s'avère être un complément intéressant aux résultats numériques finaux surtout lorsque l'allure de la fonction à minimiser au voisinage de l'optimum est très "plate".

le logiciel que nous avons mis au point s'intègre dans un ensemble cohérent d'outils robustes, fiables et bien documentés.

La partie mathématique (algorithmes de calcul ) est assurée par des programmes extraits de la librairie scientifique NAG ( Numerical Algorithms Group ).

La saisie interactive des données a été rendue conviviale grâce à un générateur de programmes d'acquisition développé en

FORTAN 77 au centre de calcul du laboratoire de Solvay.

Quant au graphisme, s'appuyant sur une norme internationale ( Graphical Kernel System ), il s'insère de manière portable dans l'ensemble.

Notre logiciel est actuellement soumis à des tests intensifs au centre de calcul de Solvay dans le but d'en éprouver la robustesse et la convivialité

Plusieurs cas d'applications industrielles sont en cours d'étude et les premiers résultats sont des plus encourageants.

Nous concluons en insistant sur le fait que l'outil qui a été développé est avant tout un outil d'aide à l'optimisation c'est-à-dire un moyen d'aider le chercheur dans des problèmes nouveaux et/ou dans des problèmes difficiles. L'introduction du graphisme gourmande en ressources répond à un besoin d'assistance; il va de soi que, lorsque les programmes accèdent à une phase d'utilisation routinière, on devrait pouvoir, sinon s'en passer, au moins en limiter l'usage.

## **Bibliographie**

# BIBLIOGRAPHIE.

- 1 BRENT, R.P. (1973) . Algorithms for minimization without derivatives. Prentice - Hall Inc. Englewood Cliffs New Jersey.
- 2 DAVIDON, W. (1959). Variable metric methods for minimization, A.E.C.Res.and Develop.Rpt. Anl. 5990 Argonne Nat'l Lab. Argonne Illinois.
- 3 J.P. DEKIMPE - Conception d'un outil portable de gestion d'écrans en FORTRAN 77.  
Mémoire de licence - décembre 85.
- 4 FLETCHER, R. and POWELL, M.J.D. (1973). On the modification of LDL<sup>T</sup> factorizations. AERE report, TP 519.
- 5 GENTLEMAN, W.M. (1973). Least squares computation by Givens transformation without square root.  
J. Inst. Math. Appl. V.12, PP 329-336.
- 6 GILL, P.E. and MURRAY, W. (1972). Quasi - Newton methods for Unconstrained optimization. J. Inst. Math. Applics 9 PP . 91 - 108.
- 7 GILL, P.E. and MURRAY, W. and PITFIELD, R.A. (1972). The Implementation of two quasi-Newton methods for unconstrained optimization. Nab. Phys. Lab.rpt NAC 11.
- 8 GILL, P.E. and MURRAY, W. (1974) Safeguarded Steplength Algorithms for Optimization Using descent methods. Nat. Phys. Lab. rpt. NAC 37.

- 9 GILL,P.E. and GOLUB,G.H. and MURRAY,W and SAUNDERS,M.A.  
(1974). Methods for modifying matrice fac-  
torization. Math. Comp. v.28. pp.505-535.
- 10 GILL,P.E. and MURRAY,W. and SAUNDERS,M.A. (1975).  
Methods for computing and Modifying the LDV  
factors of a Matrix. Math.comp.v.29.
- 11 GILL,P.E. and MURRAY,W. ( 1976 ) Minimization subjet to  
bounds on the variables. Nat. Phys. Lab.  
rpt. NAC 72.
- 12 GILL,P.E. and MURRAY,W. and WRIGHT,M.H. (1981)  
Practical optimization. Academic Press.
- 13 F.R.A. HOPGOOD, D.A. DUCE, J.R. GALLOP, D.C. SUTCLIFFE  
Introduction to the graphical kernel system  
G.K.S.  
Academic Press. 1983.
- 14 NAG CENTRAL OFFICE  
NAG FORTRAN library manuals OXFORD January 84.
- 15 J. SPELKENS et J. DG WINTER (1985) - CALCOMP - Logiciel  
Interactif de tracer graphique. Notice Solvay-  
TEC  
TEC 85 - 1001.